МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Пятигорский институт (филиал) СКФУ

# Методические указания

по выполнению лабораторных работ
по дисциплине
**«Искусственный интеллект в профессиональной сфере»**
для направления подготовки **10.03.01 Информационная безопасность**
направленность (профиль) **Безопасность компьютерных систем**

**Пятигорск**
**2022**

**СОДЕРЖАНИЕ**

# ВВЕДЕНИЕ

**Цели и задачи освоения дисциплины**

Целью освоения дисциплины является ознакомление студентов с кругом задач в области искусственного интеллекта, а также знакомство с основными понятиями и методами машинного обучения и их применением к задачам, относящимся к профессиональной области.

Задачи освоения дисциплины – ознакомление студентов с возможностями использования технологий искусственного интеллекта в профессиональной сфере.

**В результате освоения дисциплины студенты должны:**

- использовать принципы работы современных информационных технологий для решения задач профессиональной деятельности;
- выбирать цифровые технологии, программные средства для решения профессиональных задач;
- применять при решении задач профессиональной деятельности специализированное программное обеспечение и методы искусственного интеллекта;
- разрабатывать основные модули интеллектуальных систем, владеть приемами решения лабораторных задач в предметной области.

**Перечень планируемых результатов обучения по дисциплине (модулю), соотнесённых с планируемыми результатами освоения образовательной программы**

| Код, формулировка компетенции | Код, формулировка индикатора | Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов |
|---|---|---|
| **ПК-3** Способен адаптировать и модифицировать специализированное программное обеспечение, методы и алгоритмы систем искусственного интеллекта и машинного обучения в профессиональной деятельности | **ИД-1**$_{ПК-3}$ Ориентируется в современных тенденциях развития цифровых технологий, выбирает технологии или программные средства для решения поставленных задач<br>**ИД-2**$_{ПК-3}$ Применяет при решении задач профессиональной деятельности специализированное программное обеспечение и методы искусственного интеллекта | Готовность применять при решении задач профессиональной деятельности специализированное программное обеспечение и методы искусственного интеллекта.<br>Готовность разрабатывать основные модули интеллектуальных систем, владеть приемами решения практических задач в предметной области. |

## 1. Наименование лабораторных занятий

| № | Наименование тем дисциплины, их краткое содержание | Объем часов | Из них практическая подготовка, часов |
|---|---|---|---|
| 5 семестр | | | |
| 1. | Анализ современных | 3 | 3 |

| | | | |
|---|---|---|---|
| | программных средств с применением ИИ | | |
| 2. | Формализация знаний Использование семантических сетей для представления знаний | 6 | 6 |
| 3. | Создание онтологии в системе Protégé | 9 | 9 |
| 4. | Разработка учебной экспертной системы. | 3 | 3 |
| 5. | Анализ существующих образовательных платформенных решений. | 6 | 6 |
| | Итого за 5 семестр | 27 | 27 |
| | Итого | 27 | 27 |

## СТРУКТУРА И СОДЕРЖАНИЕ ЛАБОРАТОРНЫХ ЗАНЯТИЙ

### Лабораторная работа 1.

Анализ современных программных средств с применением ИИ.

**Вопросы (компетенции, навыки) для освоения:**

1. Изучить современные программные продукты, основанные на ИИ.
2. Научиться анализировать возможности программных средств с применением ИИ.

**Задания для выполнения и методические рекомендации:**

**Задание 1**

Используя приведенный ниже список интернет-ссылок, а также другие интернет-источники, изучите интеллектуальные системы, предлагаемые на рынке.

1. https://www.zeluslugi.ru/info-czentr/stati/primery-iskusstvennogo-intellekta-programmy
2. https://3dnews.ru/1058498/intellektualnoe-prevoshodstvo-10-zaslugivayushchih-vnimaniya-aiservisov-i-prilogeniy
3. https://soware.ru/categories/artificial-intelligence-platforms

Определите их направление:
- экспертные системы (ЭС);
- системы с базами знаний;
- интеллектуальное обучение;
- нейронные сети.

**Задание 2**

Проанализируйте возможности, области применения, достоинства и недостатки изученных интеллектуальных систем. На основе анализа составьте отчет в табличной форме:

| Название ПО или сервиса, версия | Возможности программы | Область применения | Стоимость | Источник информации (www. ) |
|---|---|---|---|---|
| | | | | |

Сделайте вывод о рынке ПО. Рассчитайте удельный вес по области применения ПО. Изобразите результат графически (постройте диаграмму).

## Лабораторная работа 2
## Формализация знаний
## Использование семантических сетей для представления знаний

**Вопросы (компетенции, навыки) для освоения:**

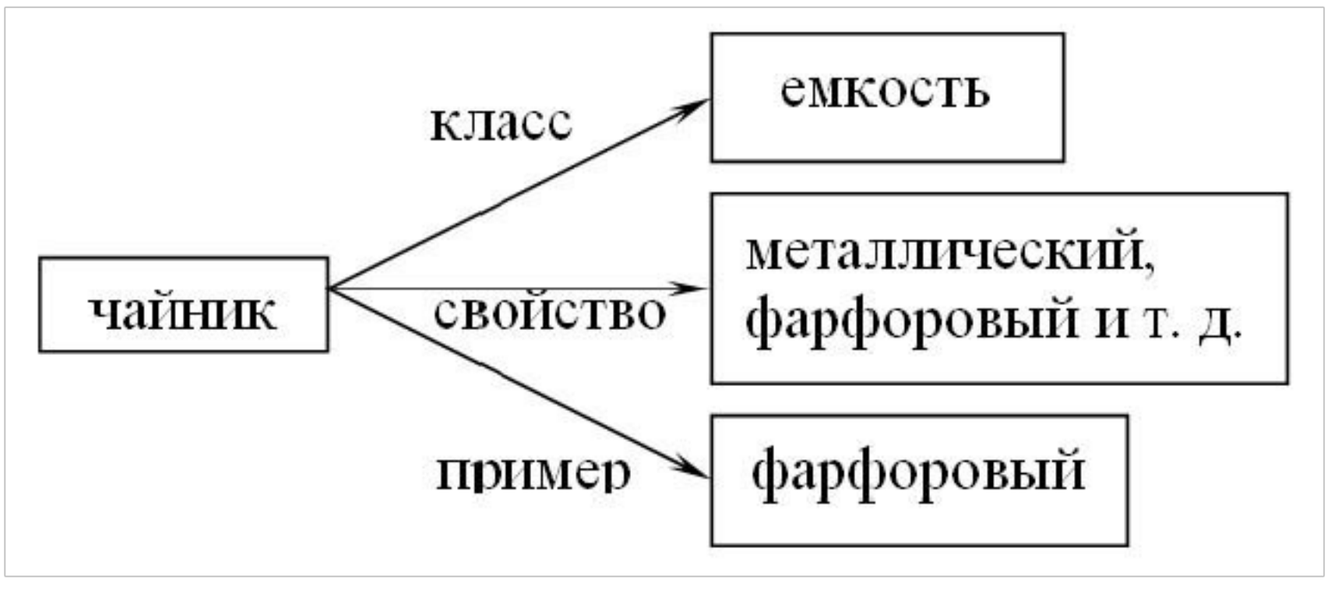
1. Изучить общее представление о семантической сети и фреймах для представления знаний.

2. Научиться строить семантическую сеть и используя фреймы реализовать структуру отношений.

**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.

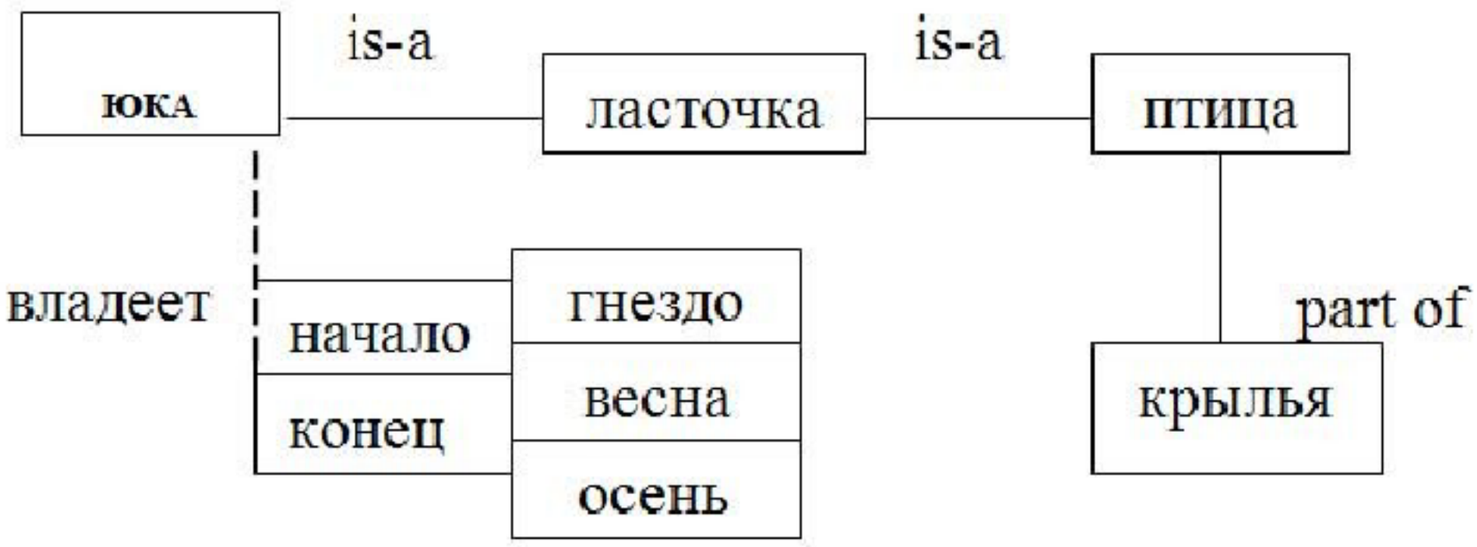



Простейший образец семантической сети

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 (является) и PART OF (является частью) отношение:

Достоин IS – A человека, «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 иерархическая структура понятия ЮКО. 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 в слотах 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.

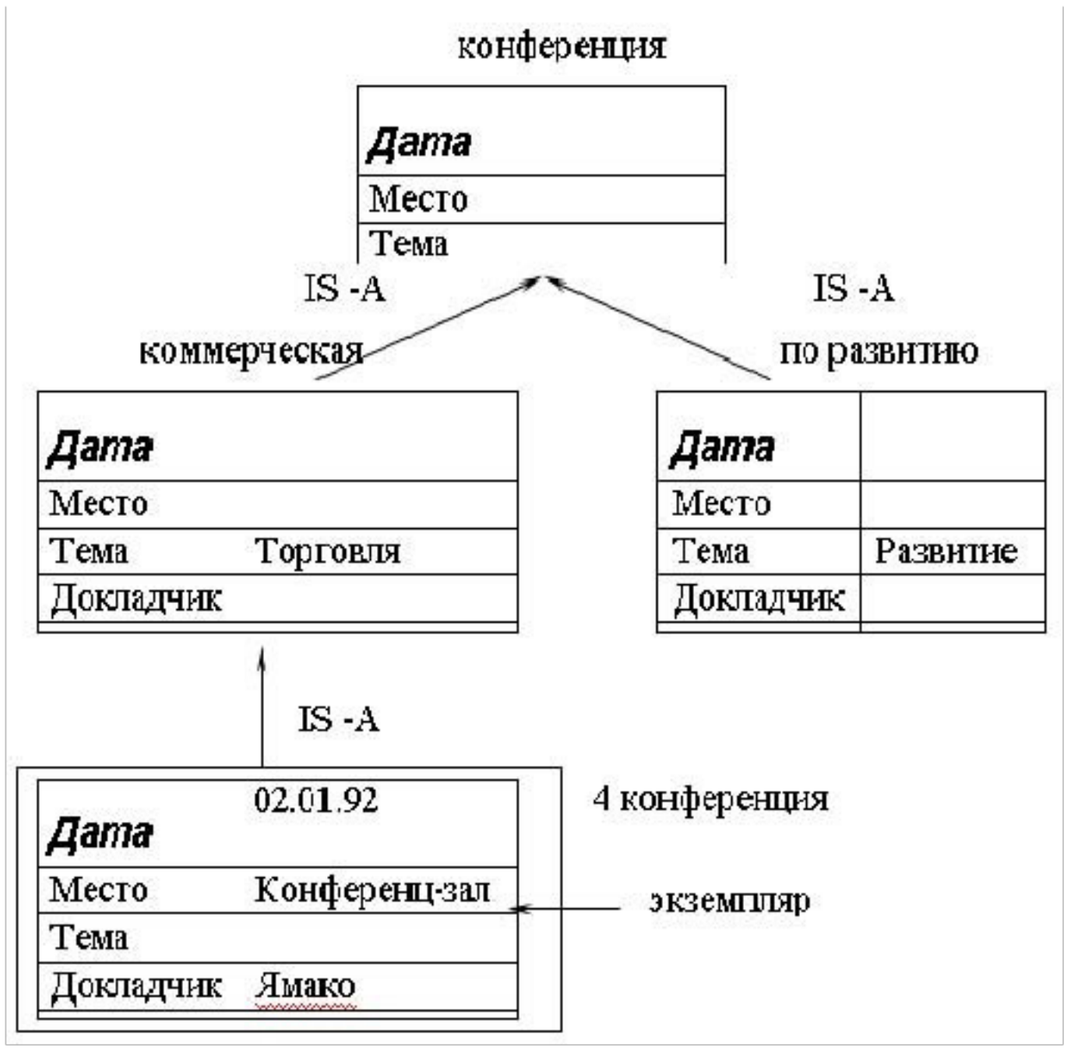
7. Присоединенная процедура. В качестве значения слота можно использовать программу процедурного типа. Когда мы говорим, что в моделях представления знаний фреймами объединяются процедурные и декларативные знания, то считаем демоны и присоединенные процедуры процедурными знаниями.

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.

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.




Пример фреймовой модели

**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 оборудование.

9. животный мир при составляющих: вид, тип, среда обитания, особенности поведения.

## Лабораторные работы 3-5

## Создание онтологии в системе PROTÉGÉ. Основы RDF и OWL

**Вопросы (компетенции, навыки) для освоения:**

3. Определить предметную область, создать классы, отношения с использованием системы Protégé, а также на примере созданной БЗ рассмотреть модель RDF, язык OWL и синтаксис Turtle.
4. Научиться реализовывать понятия Defined Classes, DL Query в системе Protégé
5. Изучить возможности языка запросов SPARQL
6. Научиться анализировать создавать правила для проверки информации о товарах с помощью языков SWRL и SQWRL .

**Задания для выполнения и методические рекомендации:**

В качестве примера будет рассмотрена информация о конкретном товаре (конкретном смартфоне[1]) и на ее основании составлена первая версия БЗ.

Перед началом работы с системой Protégé [6] необходимо составить хотя бы некоторую упрощенную концептуальную модель предметной области. И, для того чтобы потом не было сложностей с ее формализацией, необходимо определить какими сущностями можно оперировать. В данном случае это сущности модели RDF [1, 2] и языка OWL [4], т.к. многие системы и библиотеки, в том числе Protégé поддерживает эти стандарты, и они являются наиболее распространенными.

Стоит сразу отметить, что модель RDF является абстрактной и существует множество языков с разными синтаксисами, реализующих ее [1]. OWL также может быть преобразован в несколько нотаций и в любой из языков RDF [4] Конкретные синтаксисы будут обсуждаться позднее и все



[1] https://www.dns-shop.ru/product/052629b3621f3330/62-smartfon-samsung-galaxy-s9-256-gb-fioletovyj/characteristics/

примеры, описанные ниже могут их не придерживаться.

Онтология в рамках модели RDF предполагает определение любых понятий предметной области в виде троек: субъект – предикат – объект. Множество таких троек образует граф понятий. Субъекты и объекты интерпретируются как вершины ориентированного графа, а предикаты как его дуги (и определяют отношения между сущностями). Вершины могут быть следующих типов: ресурс, пустая вершина.

Ресурс описывает некоторое понятие реального мира и разделяется на ссылочные ресурсы и скалярные. Ссылочные ресурсы – это некоторые понятия, которые могут иметь характеристики (например, смартфон, смартфон SMG965FZPHZER, экран). Таким понятиям присваивается глобальный уникальный идентификатор IRI (Internationalized Resource Identifier) (например, http://myontololgy.org/goods#phone1). Соответственно, такие ресурсы могут находиться как на месте субъекта, так и на месте объекта в зависимости от предиката.

Скалярные ресурсы указывают некоторые конкретные характеристики и являются просто значением определенного типа. Соответственно, такие ресурсы могут находится только на месте объекта. Но, при этом для них может определяться тип и для строковых значений может определяться язык. Например, для нашей предметной области такими сущностями могут являться: 6.2 (конкретная диагональ экрана; float), «Samsung Galaxy S9+¬ (оригинальное название; string); «черный¬@ru (цвет; string; русский).

Также необходимо отметить, что «предикат¬ также может быть «субъектом¬ или «объектом¬ в рамках других троек. Это необходимо чтобы, например, описать ограничения данного отношения.

Пустая вершина в основном необходима для «имитация¬ n-арных связей. Как видно из концепции модели, предикат может связывать только 2 ресурса. Но, это решается с помощью введения промежуточной вершины и разделения на несколько бинарных. При этом вершина не будет представлять понятие реального мира и ей нет смысла давать уникальный

глобальный идентификатор, достаточно локального для составления троек. Именно для таких вершин и введен обсуждаемый тип. При этом не всегда очевидно стоит ли вводить для той или иной сущности глобальной идентификатор. Например, конкретный смартфон обладает рядом характеристик дисплея и было решено выделить дисплей как отдельную сущность. При этом для нас он является неотъемлемой частью смартфона, а производитель смартфона может ставить разные модели дисплея от разных производителей при условии сохранения характеристик. Тогда может быть целесообразно использовать пустую вершину. С другой стороны, если производитель присвоил данному дисплею определенный код, то может быть целесообразно выделить его как ресурс. Т.к. он может быть установлен и на другой смартфон, а также будет проще реализовывать, например, запросы для определения сервисов, где есть в наличии данный дисплей. Для примера пустой вершины рассмотрим характеристику «разрешение¬ у конкретного дисплея:

SM-D-G965FZPHSER разрешение _x

_x горизонталь 2960

_x вертикаль 1440

Данные тройки описывают граф, изображенный на рис. 1.

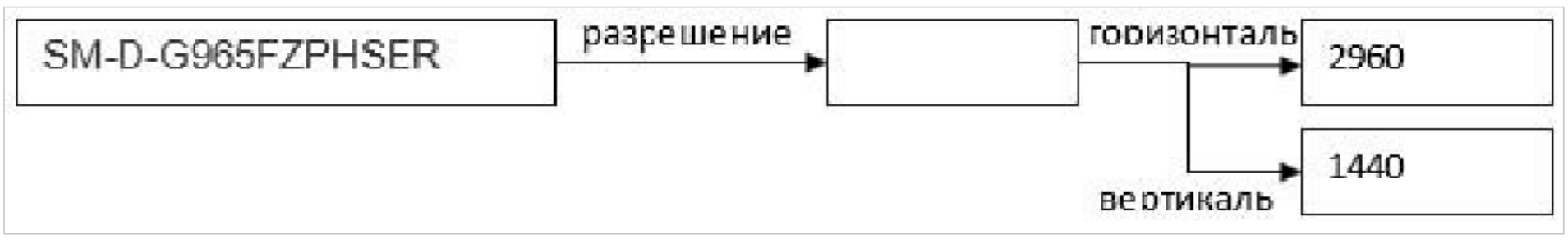



Рис. 1 Граф, описывающий разрешение дисплея

Стоит отметить, что данные понятия также можно было описать с использование 2 связей «разрешение_гор¬, «разрешение_вер¬ и без промежуточной вершины. Что показывает, что создание онтологии остается вариативным процессом, зависящим от конкретных целей и предпочтений разработчика.

Как правило, IRI ресурса состоит из идентификатора БЗ или

пространства имен и идентификатора понятия в нем. Например, «http://myontololgy.org/goods#phone1¬, может означать, что описывается понятие «phone1¬ в БЗ товаров с идентификатором «http://myontololgy.org/goods#¬. Чтобы не писать каждый раз полный идентификатор БЗ или пространства имен, его заменяют на префикс. Например, можно записывать «mo:phone1¬, где под «mo:¬ будет подразумеваться полный IRI БЗ.

Стандарт RDF вводит не только концепцию, но и 3 пространства имен с общими понятиями: rdf, rdfs, xsd [3]. Где xsd описывает типы, которые можно использовать в онтологиях на базе RDF (например, xsd:string, xsd:integer, xsd:double, xsd:boolean). Например, разрешение дисплея по горизонтали могло быть описано следующим образом: _x горизонталь “2960”^^xsd:integer. Основные понятия из rdf, rdfs, которые будут использоваться в рамках данной работы: rdfs:Resource (понятие ресурса, описанного ранее), rdfs:Class (понятие класса), rdf:Property (понятие отношения), rdfs:subClassOf (отношение, которое определяет «субъект¬ как подкласс «объекта¬), rdfs:subPropertyOf (отношение, которое определяет «субъект¬ как уточнение «объекта¬, являющегося отношением), rdfs:domain (отношение, определяющее область определения другого отношения), rdfs:range (отношение, определяющее область значения другого отношения), rdf:type (отношение, которое определяет «субъект¬ как экземпляр «объекта¬).

OWL является стандартизированным языком создания онтологий. Он имеет несколько вариантов синтаксиса (в том числе функциональный), а также может быть преобразован в любой из синтаксисов RDF. Т.к. OWL совместим с RDF, то в нашем случае он будет интерпретироваться как надстройка над RDF, которая уточняет подход к созданию онтологий, расширяет возможности модели и вводит новые сущности (из пространства имен owl).



БЗ, созданные с использованием OWL предполагают, что на них

производится вывод, который позволяет проверить непротиворечивость БЗ, вывести новые отношения.

В связи с этим описание онтологии можно разделить на:

1 Сущности. Все понятия, описанные ранее в рамках RDF, а также некоторые новые из OWL;

2 Выражения. Описание понятия через комбинацию множества других сущностей. Одним из основных примеров применения является определение класса через выражение. Например, «Камерофон¬ – это телефон, который имеет не менее 2 основных камер с разрешением не менее 20мегапикселей; новый тип «число¬ – это объединение множеств целых чисел и чисел с дробной частью.

3 Аксиомы. Точно известные положения и положения, которые всегда должны оставаться истинными. Например, на отношение «плотностьПикселей¬ могут быть наложены ограничения на область определения – экземпляр класс «Дисплей¬ и область значения – «целое число¬. При этом если в базе знаний будет тройка «G плотностьПикселей 529¬, то это может привести как к ошибке (если G точно не может являться экземпляром класса «Дисплей¬), так и к выводу нового отношения «G rdf:type Дисплей¬.

Как уже было сказано выше, OWL добавляет новые понятия и аксиомы. Для данной работы необходимы следующие понятия:

1 owl:Class. Аналогично rdfs:Class;

2 owl:Thing. Класс, являющийся предком всех классов. Аналогичен Object в некоторых языках программирования;

3 owl:ObjectProperty. Подкласс отношений (rdf:Property), связывающих два ресурса с IRI;

4 owl:DataProperty. Подкласс отношений (rdf:Property), связывающих ресурс с IRI и литерал;

5 owl:TopObjectProperty/owl:TopDataProperty. Верхнее в иерархии object/data property, являющееся предком всех object/data property;

6 owl:FunctionalProperty. Класс отношений, применяемых к одному «субъекту¬ только один раз. Т.е. в онтологии не может существовать два триплета «x c y1¬, «x c y2¬, где y1 и y2 разные сущности и «c rdf:type owl:FunctionalProperty¬;

7 owl:disjointWith. Отношение, декларирующее, что экземпляр субъекта-класса не может быть одновременно экземпляром объекта-класса. Т.е. в онтологии не может существовать два триплета «x rdf:type y1¬, «x rdf:type y2¬, где y1 и y2 разные классы и «y1 owl:disjointWith y2¬.

8 Сущности, необходимые для составления выражений. Будут описывать при появлении (owl:members,, owl:unionOf и т.д.)

Теперь, когда сущности, которые могут быть использованы, обозначены, можно спланировать общее описание предметной области. Оно разделено на описание классов, экземпляров и отношений между ними (рис. 2) и описание самих отношений (рис. 3).

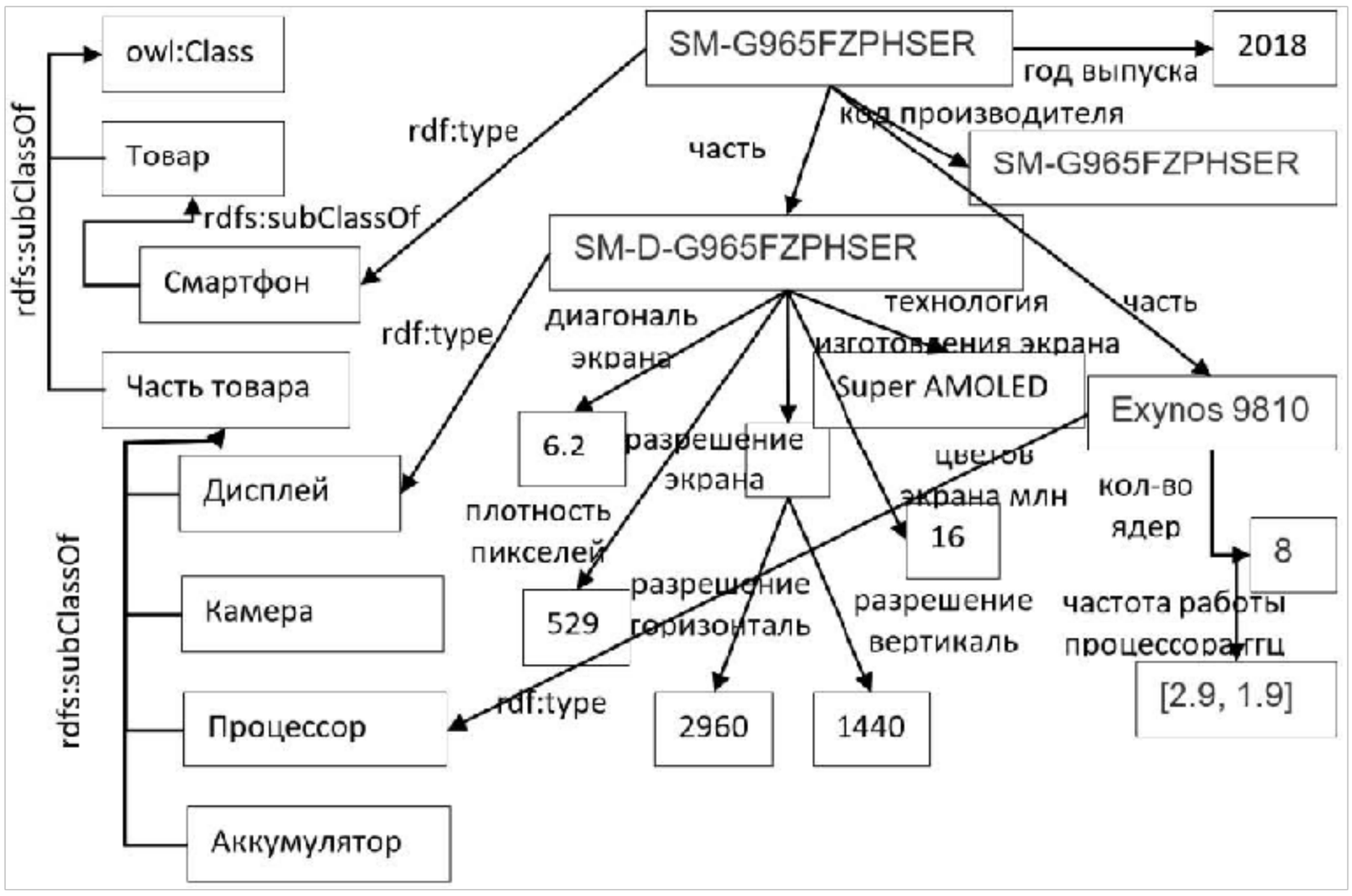



Рис. 2 Фрагмент графа, описывающего классы, экземпляры и их отношения в рамках описания смартфонов

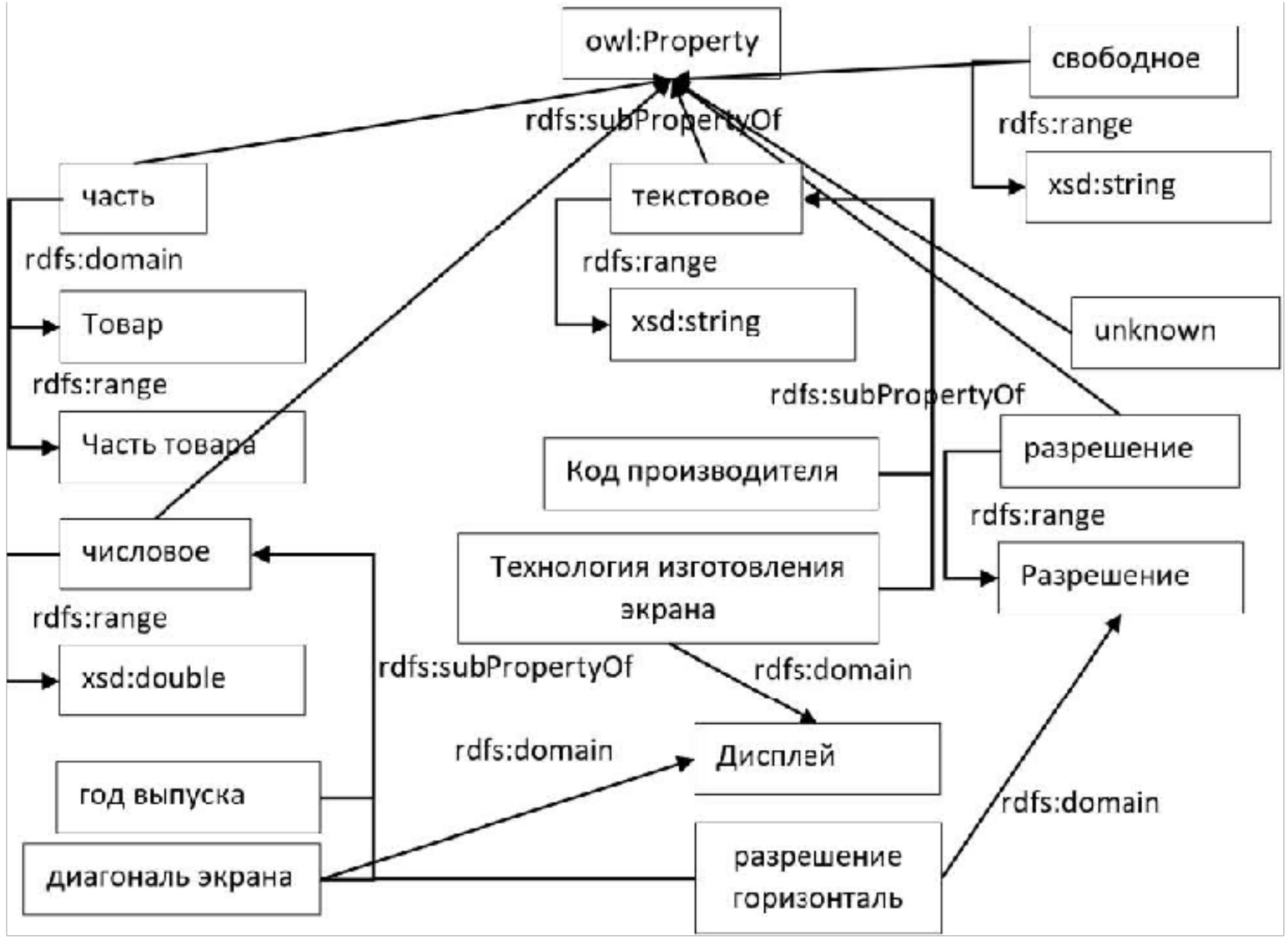



Рис. 3 Фрагмент графа, описывающего свойства в рамках описания смартфонов

В рамках приведенного фрагмента классов выделены очевидные классы:

«Товар», «Смартфон». Также выделены некоторые отдельные составляющие смартфона как подклассы класса «Часть товара»: «Дисплей», «Камера», «Процессор», «Аккумулятор». Сюда выделены только самые важные части, которые больше всего интересуют покупателей и наиболее ценны при обработке информации о товаре. Остальные характеристики будут задаваться отношениями между конкретным смартфоном и конкретными значениями (на изображении в качестве примера таких отношений можно видеть: «год выпуска», «код производителя»). Стоит отметить, что некоторые узлы (например, «[2.9, 1.9]») заданы в абстрактном виде и их конкретное задание будет зависеть от используемого синтаксиса и/или возможностей инструментальной системы.

В рамках приведенного фрагмента свойств не сделано разделение на ObjectProperty и DatatypeProperty во избежание излишней конкретизации (при реализации разделение делается очевидным образом). В рамках прямых

наследников «owl:Property» выделены типы возможных отношений: «часть» (для связи товара с его частью), «числовое/текстовое/свободное/разрешение» (данные свойства определяют группу свойств, объединенных общей областью определения. Разделение на текстовое и свободное сделано для упрощения дальнейшей обработки. Под текстовыми подразумеваются конкретные значения, которые потом могут быть преобразованы в перечисляемый тип. В рамках свободных же описания могут содержать несколько характеристик, одну характеристику, записанную разным образом и т.д.), «unknown» (группа характеристик, добавляемых автоматически в процессе сбора информации, для которых невозможно определить принадлежность к ранее описанным группам).

Необходимо понимать, что в рамках рис. 2, рис. 3 приведены только фрагменты графа для упрощения. Оставшиеся фрагменты описываются аналогично и не обладают дополнительными особенностями.

Т.к. описание должно подразумевать возможность сбора информации с использованием программных средств, необходимо обговорить как при этом будет происходить добавление новых сущностей:

1. Получить с детальной страницы товара список характеристик в виде «название
   – текст» и других необходимых отдельных полей (ссылка, код товара, категория и т.д.);

2 Произвести поиск подкласса класса «Товар», соответствующего категории рассматриваемого товара. Если не найден, то создать;

3 Создать экземпляр полученного в п.2 класса (локальная часть IRI формируется как «Код производителя» или как «Код товара» при его отсутствии)

4 Для каждой характеристики

4.1. Проверить, относится ли характеристика к описания какого-то из подклассов класса «Часть товара». Подклассы класса «Часть товара», а также отношения, связанные с ними заданы заранее и не изменяются в

процессе сбора информации. Для каждого из таких классов задаются списки полей и правила их преобразования. Конкретные реализации этой части не будут рассматриваться.

4.2. Если относится, то получить экземпляр (а при отсутствии создать) соответствующей части (формирования локальной части IRI для таких частей также прописано в правилах из п. 4.1.). Для полученного экземпляра задать отношение, связывающего его с рассматриваемой характеристикой.

4.3. Если не относится, то найти отношение по имени характеристики (при отсутствии создать отношение с таким именем как дочернее свойство для свойства «unknown» или типизированного свойства). Связать экземпляр товара с рассматриваемым значением характеристики полученным отношением.

Здесь проявляется гибкость онтологического подхода. В отличии от реляционной модели, в любой момент легко могут быть добавлены новые отношения, произведен перенос их в другую группу, произведено задание им новых ограничений. Также работа с описанием предметной области в целом ведется в терминах объектной или любой интересующей нас модели. При этом изменения, как правило, не ведут к долгому ожидания (т.к. чаще всего достаточно изменения нескольких троек).

Получив общее неформальное описание модели, можно приступать к ее формализации и реализации БЗ с использованием конкретного синтаксиса и/или инструментария. Эти процессы будут делаться параллельно с использованием системы Protégé.

Первоначально после запуска системы необходимо задать идентификатор онтологии на вкладке «Active ontology» в поле «Ontology IRI» (рис. 3б). Данный идентификатор также определяет и базовый префикс. Для изображенной онтологии IRI будет «http://myontologies.com/goods» а базовый префикс для всех создаваемых сущностей «http://myontologies.com/goods#»

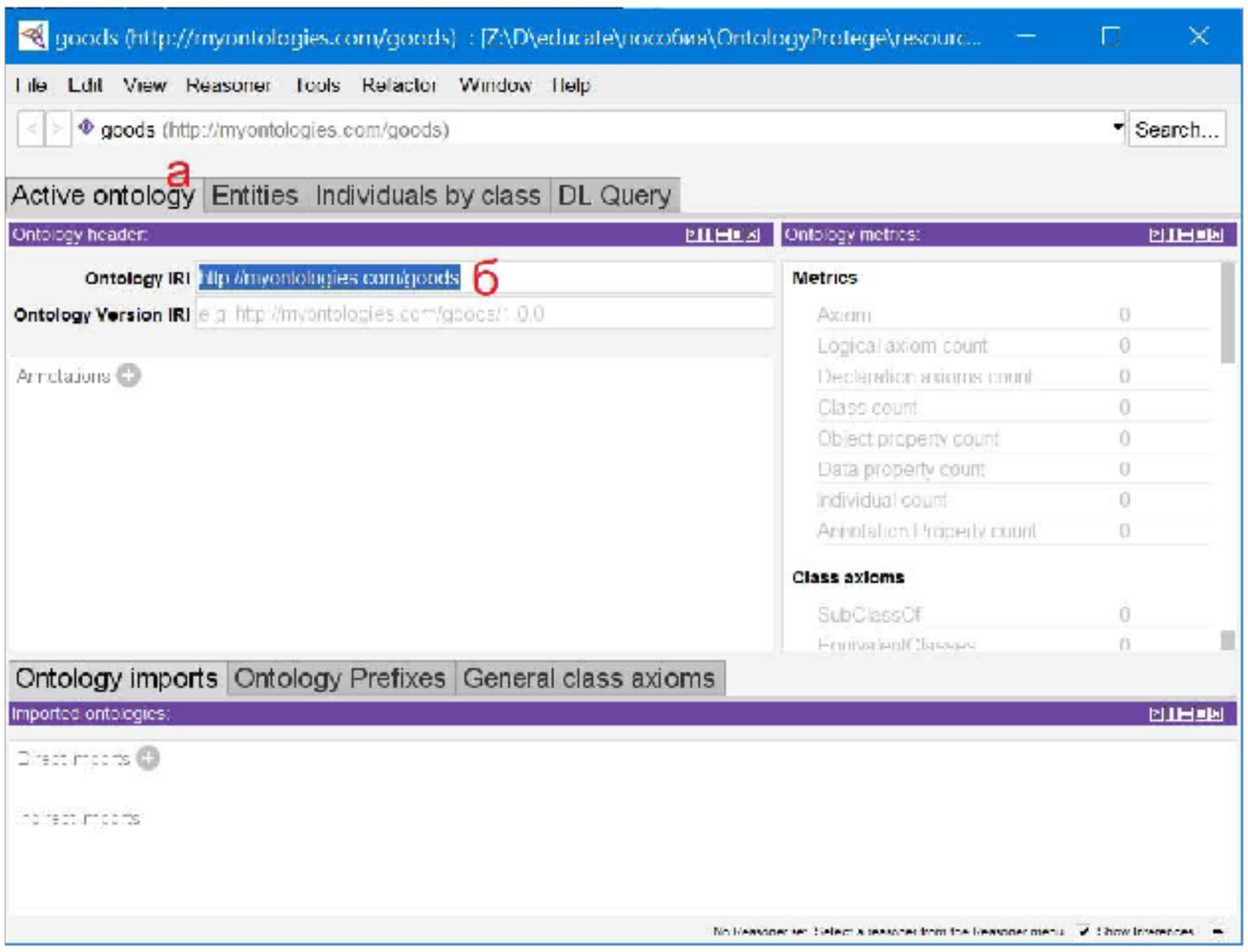



Рис. 4 Вкладка «Active ontology» в системе Protégé (а – панель в вкладок с активной вкладкой «Active ontology»; б – идентификатор онтологии)

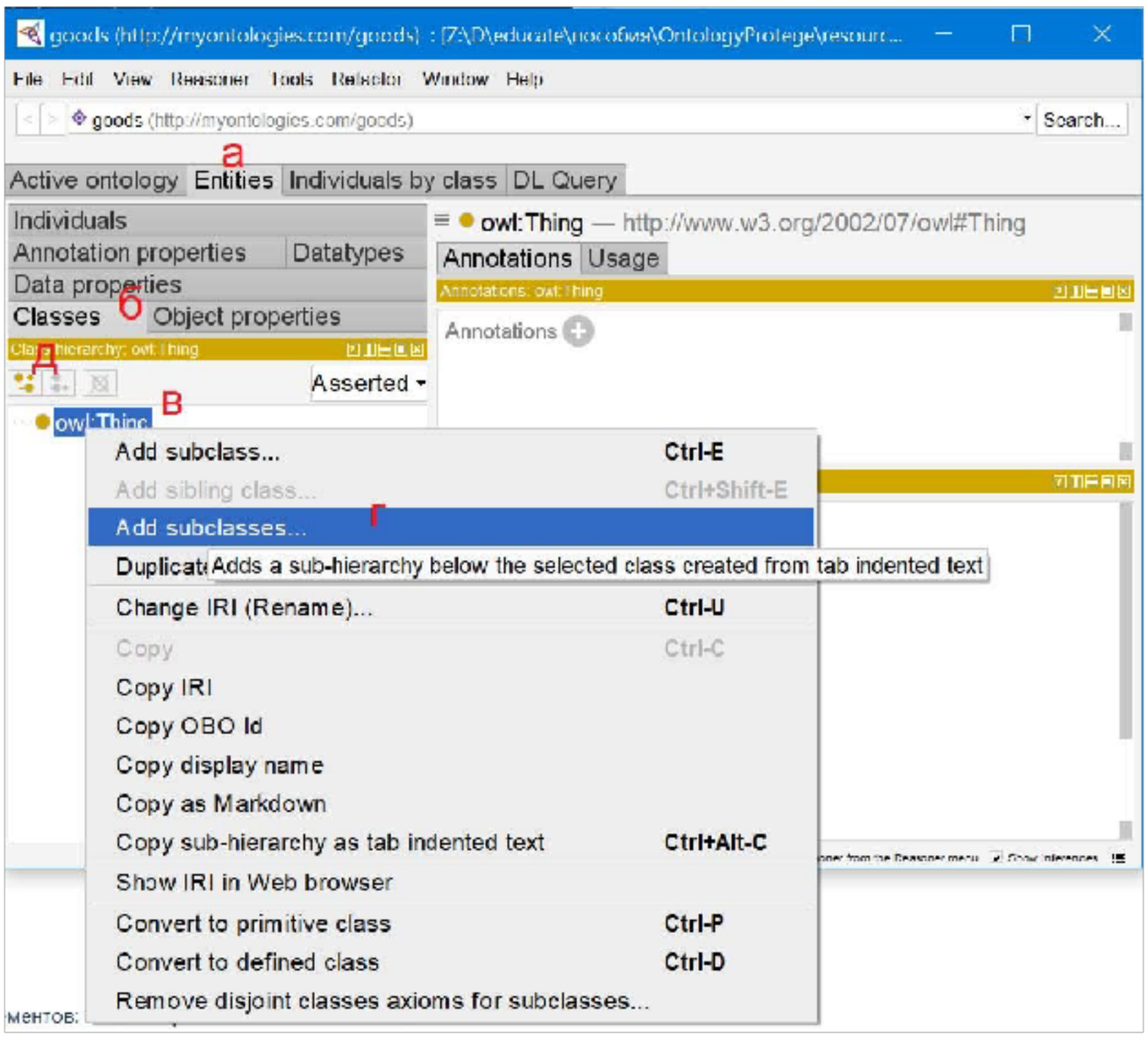



Рис. 5 Вкладка «Entities» в системе Protégé (а – панель в вкладок с активной вкладкой «Enitites»; б – панель вложенных вкладок вкладки «Entities» с активной вкладкой «Classes»; в – суперкласс owl:Thing; г, д – кнопки добавления иерархии подклассов/подкласса для выделенного класса)



Далее необходимо создать иерархию классов. Для этого необходимо выделить класс, подклассами которого будут верхние классы нашей

иерархии (в нашем случае owl:Thing, рис. 5в). После этого вызывать контекстное меню (правой кнопкой мыши – ПКМ) и выбрать пункт «Add subclasses» (рис. 5г). Классы также могут добавляться по одному (рис. 5д).

При нажатии на кнопку «Add subclasses…» появляется окно добавления иерархии подклассов (рис. 6). Каждая введенная строка в данном окне создает отдельный класс (путем добавления в онтологию троек «x rdf:type owl:Class . x rdfs:subClassOf y»). Где x – IRI создаваемого класса, задаваемый как «{IRI онтологии}#{обработанное_имя_класса}» (в рамках обработки имени класса, происходит, например, замена пробела на «_»);y – IRI родительского класса, который задается как owl:Thing для классов, заданных без отступа и как IRI ближайшего верхнего класса с меньшим отступом для остальных (отступы задается как «Tab»).

После нажатия на «Continue» появляется окно, предлагающее создать для всех создаваемых классов одного уровня отношения «owl:disjointWith». Для этой задачи соглашаемся.

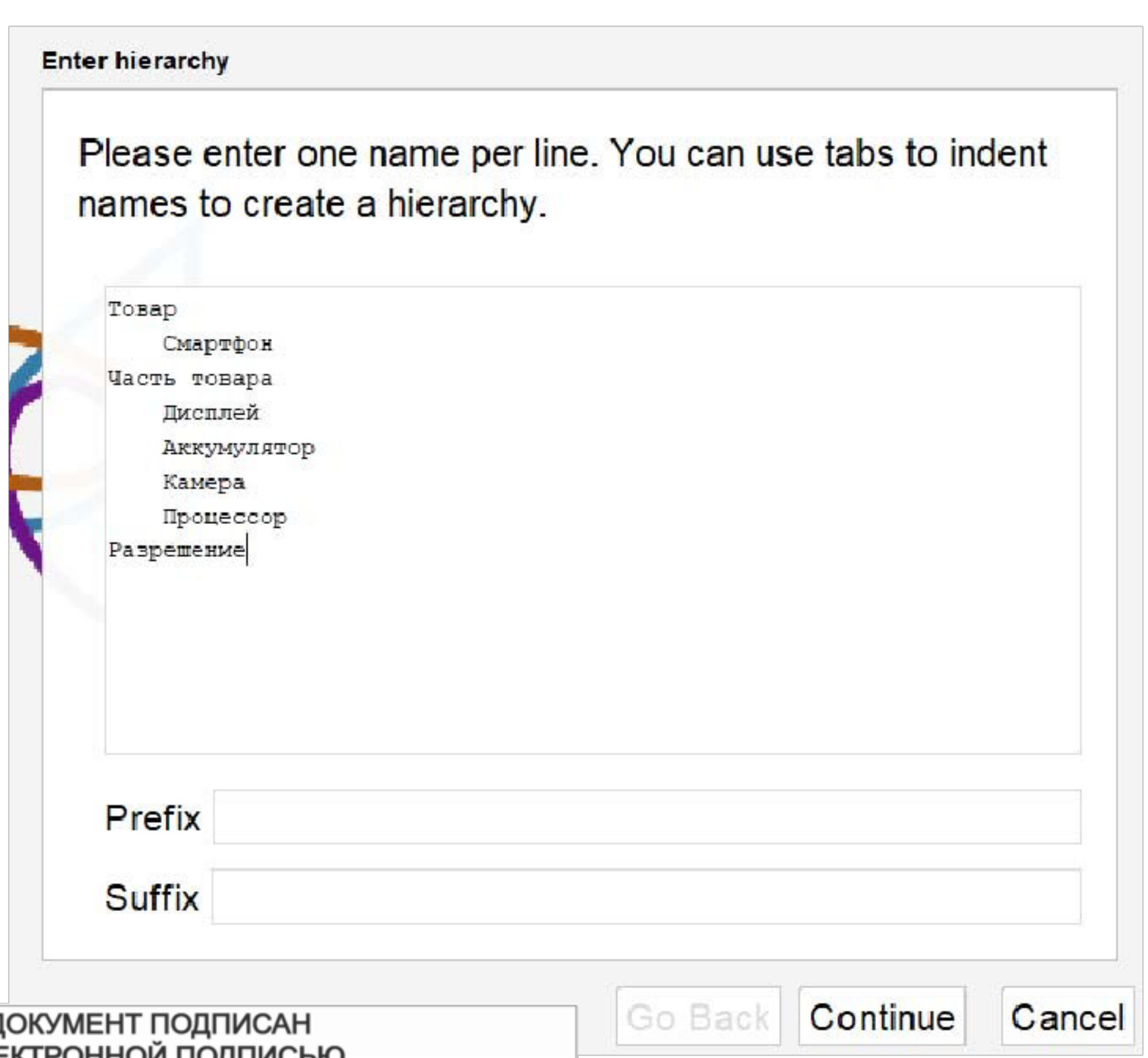



Рис. 6 Окно создания иерархии подклассов в системе Protégé

Пока БЗ является достаточно простой, рассмотрим ее графическое представление и один из синтаксисов OWL/RDF. Для вывода графического представления необходимо активировать вкладку «Window -> Tabs -> OntoGraf» и перейти на нее (рис. 7). Здесь будут видны классы, экземпляры и object property. Рассмотрение текстового представления позволит нам увидеть какие именно тройки инструментальная система создала на данный момент и лучше понять как она работает (в некоторых случаях поведение системы может быть настроено, здесь рассматривается поведение по умолчанию). Для этого необходимо сохранить онтологию (File -> Save As). При сохранении на данном этапе необходимо выбрать синтаксис «Turtle Syntax».

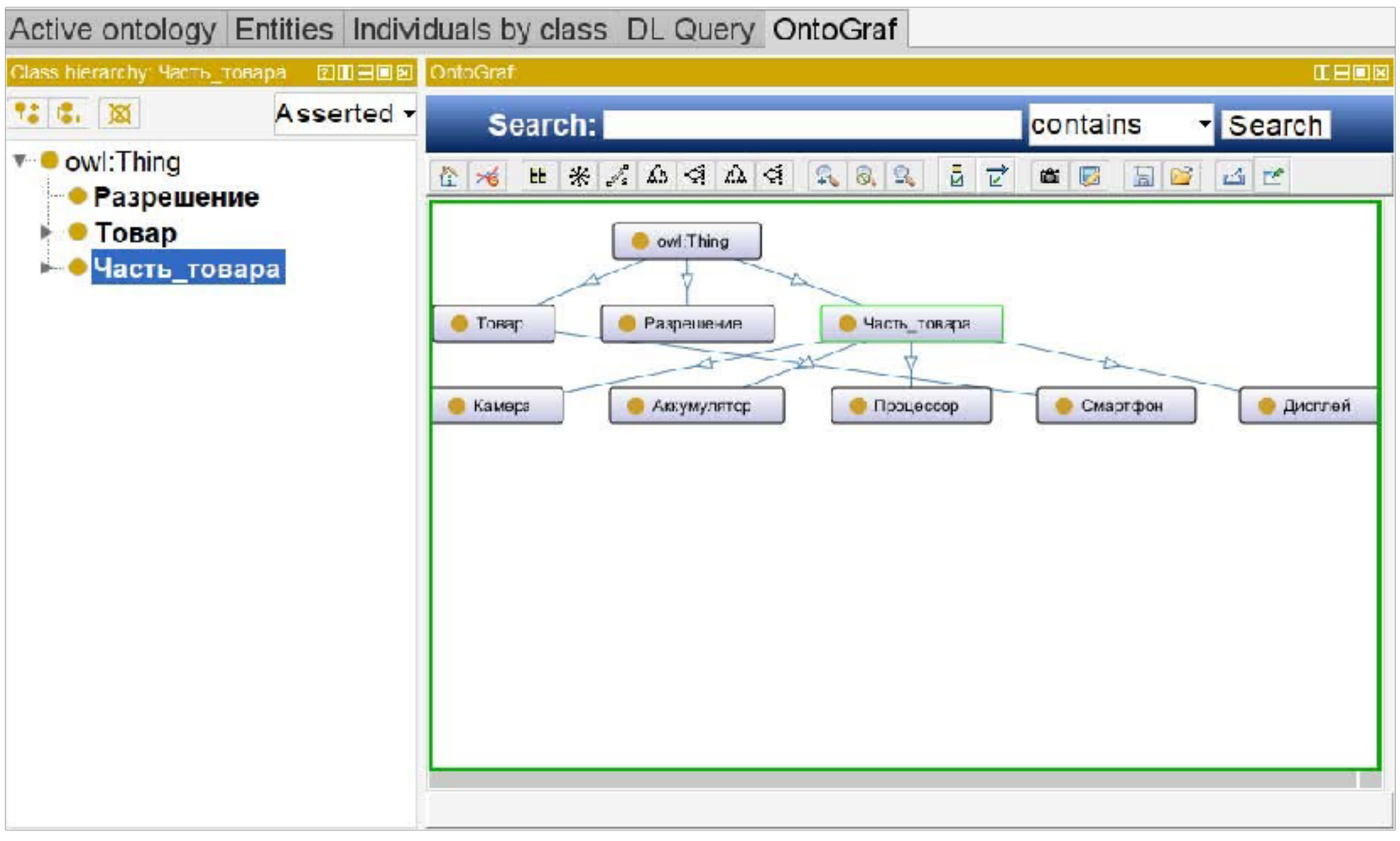



Рис. 7 Вкладка «OntoGraf» в системе Protégé

Открыв получившийся файл можно увидеть приведенный далее текст (приведены фрагменты с комментариями, объясняющими синтаксис и общую структуру онтологии. Комментарии начинаются с #).

#Описание начинается с объявления префиксов

#Пространство имен по умолчанию

можно вместо него писать «:»

@prefix : <http://myontologies.com/goods#> .

#Определение префиксов owl, rdf, xml, xsd, rdfs

# «.» является символом разделителем троек

@prefix owl: <http://www.w3.org/2002/07/owl#> .

...

#далее IRI нашей онтологии определяется как экземпляр онтологии

<http://myontologies.com/goods> rdf:type owl:Ontology .

#Далее следует описание созданных классов

#«;» является специальным разделителем троек и после него идет двойка «предикат – объект», а «субъект» берется из предыдущей тройки

:Аккумулятор rdf:type owl:Class ;

rdfs:subClassOf :Часть_товара .

#Т.е. это эквивалентно записи «:Аккумулятор rdf:type owl:Class . :Аккумулятор rdfs:subClassOf :Часть_товара»

#Для подклассов owl:Thing отношение rdfs:subClassOf может быть опущено при генерации файла

:Разрешение rdf:type owl:Class .

...

#Далее описываются общие аксиомы. В нашем случае это несколько условий на группы классов одного уровня, что никакие сущности не могут быть экземплярами одновременно нескольких классов из группы

[ rdf:type owl:AllDisjointClasses ;

owl:members ( :Аккумулятор :Дисплей :Камера :Процессор )] .

Преобразуем последнюю запись в вид троек для ее понимания, учитывая что: «[]» – позволяет задавать пустые узлы; «()» – позволяет задавать списки (для задания списков в RDF есть специальный класс rdf:List и отношения rdf:first, rdf:rest):

_x rdf:type owl:AllDisjointClasses .

_x owl:members _y .

y rdf:type rdf:List . _y rdf:first :Аккумулятор . _y rdf:rest _e .

_e rdf:type rdf:List . _e rdf:first :Дисплей . _y rdf:rest _l .

_l rdf:type rdf:List . _l rdf:first :Камера . _y rdf:rest _k .

_k rdf:type rdf:List . _k rdf:first :Процессор . _y rdf:rest rdf:nill .

Это также может быть записано в функционально стиле:

DisjointClasses(

:Аккумулятор :Дисплей :Камера :Процессор )

Понимание стандартов и синтаксиса облегчает дальнейшую работу с онтологией (в особенности работу с запросами и редактирование онтологии из программного кода). Кроме того, система Protégé реализует на уровне интерфейса не все операции (в частности, нет возможности явного создания пустых узлов, но при этом система в состоянии их понять в большинстве случаев). Также стоит отметить, что многие описания так или иначе придется писать самостоятельно в соответствии с определенным синтаксисом (defined classes, сложные ограничения на поля и т.д. реализуются в системе с использованием Manchester Syntax [7]).

Далее необходимо задать Object Properties. Они задаются аналогично иерархии классов на вкладке «Object Properties». (рис. 8)

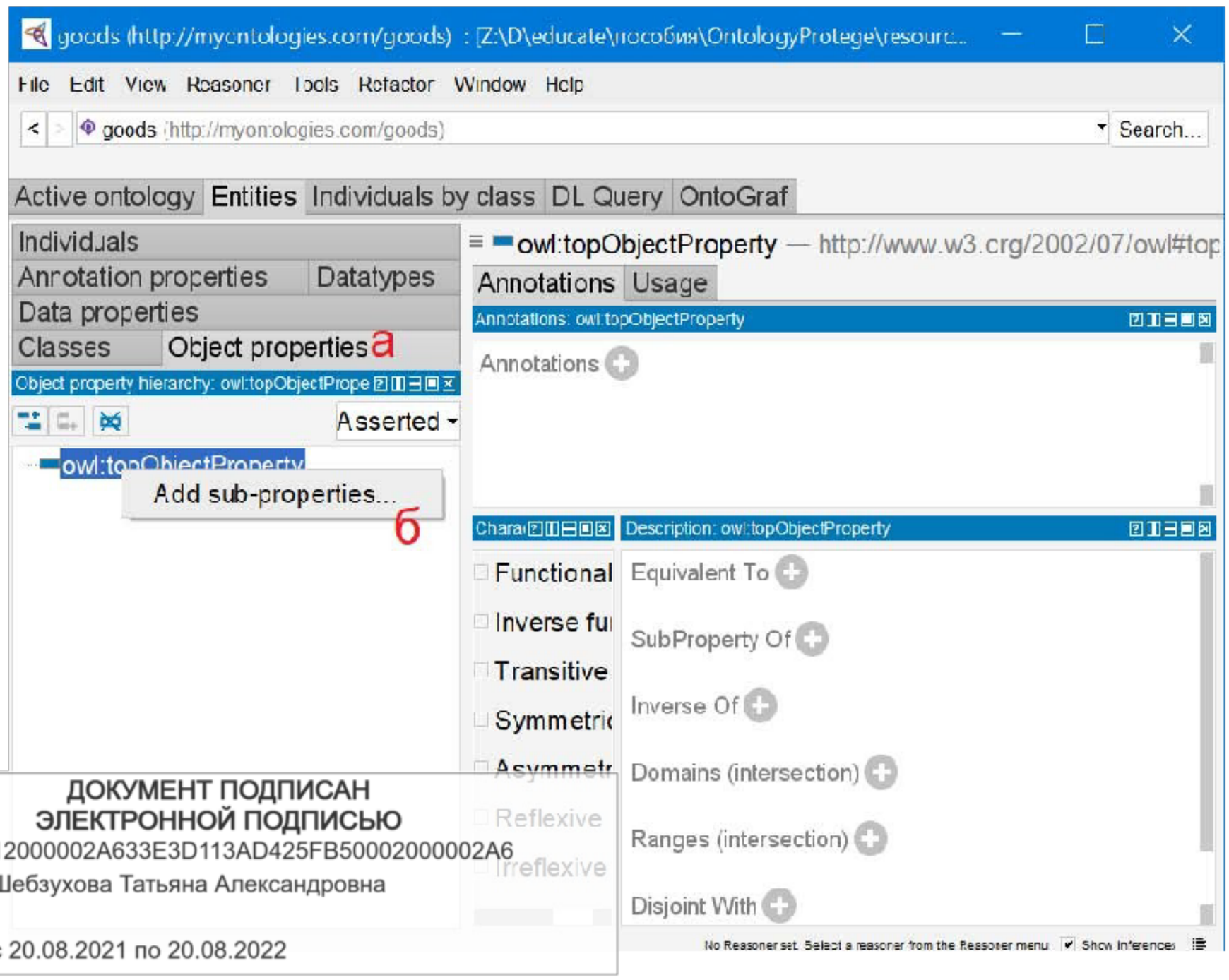


Рис. 8 Добавление Object Properties (а – активная вкладка Object Properties, б– кнопка добавления иерархии отношений)

На данном этапе необходимо задать свойства «часть» и «разрешение». Далее опишем эти свойства подробнее, а именно зададим область определения и область значения.

Для свойства «часть» область определения «Товар», область значения «Часть_товара». Для того, чтобы задать эту информацию, необходимо выбрать свойство (рис. 9а), найти раздел «Domains» в появившемся описании и добавить новую запись (рис. 9б), выбрав соответствующий класс сущностей (рис. 9г).

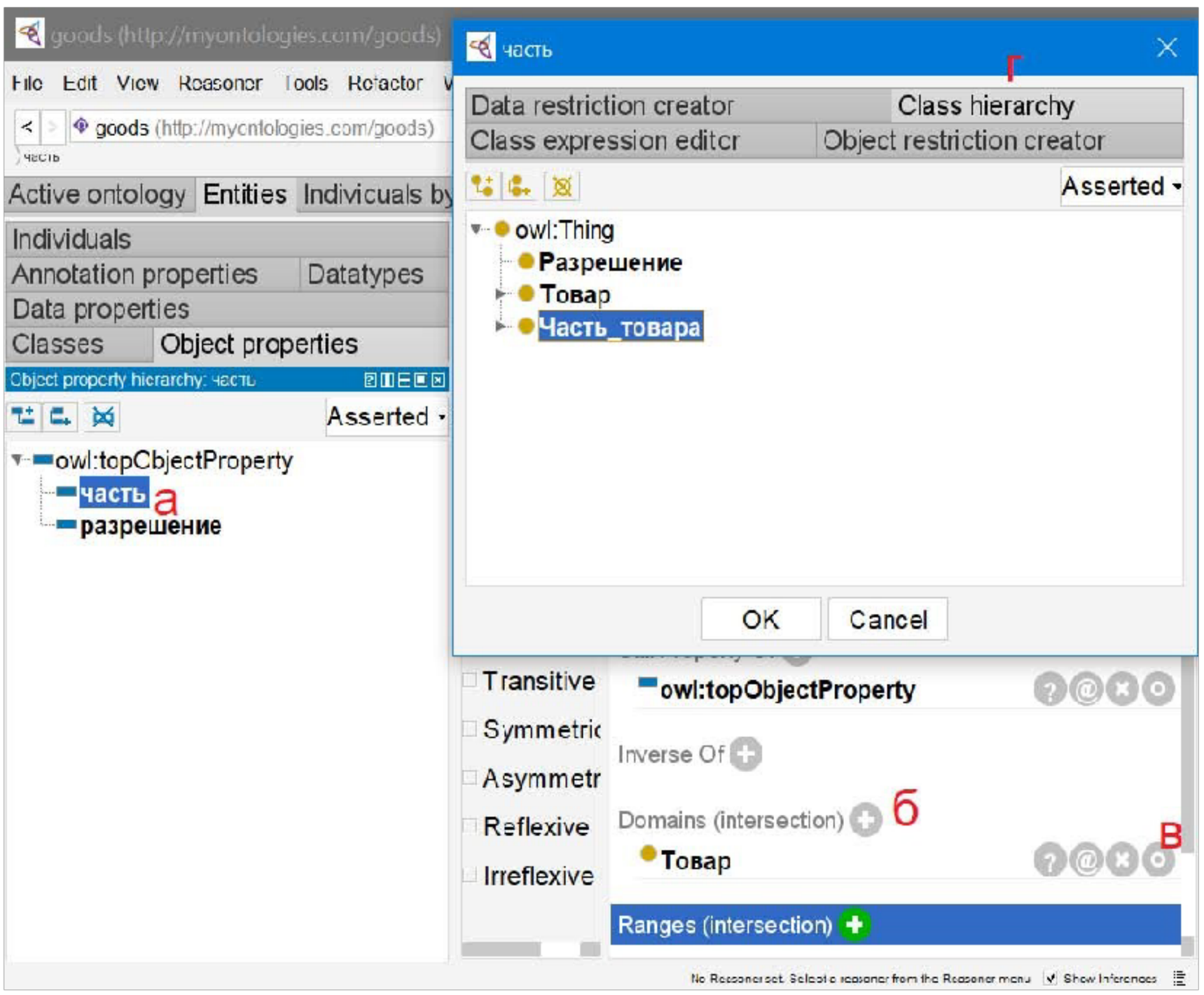



Рис. 9 Задание информации о Object Property в системе Protégé (а – выделенное свойство «часть», б – кнопка добавления новой области определения, в – кнопка редактирования существующей области определения, г – активная вкладка иерархии классов при задании области определения свойства)

При этом возможен не только выбор из существующих классов, но и задание через выражения на вкладках «Class expression editor», «Object restriction creator» (они являются одним из примеров, где придется использовать Manchester Syntax, а не просто графический интерфейс).

По аналогии создаются Data Properties. Полный список свойств, создаваемых в данной БЗ, приведен ниже (при создании необходимо заменить «;» на переносы строк и отступ)

текстовое

код производителя; цвет заявленный производителем; сеть 2g; сеть 3g; частота lte; формат sim карты; технология изготовления экрана; материал корпуса; защита корпуса; защита экрана; степень защиты; операционная система; производитель процессора; модель процессора; конфигурация процессора; графический ускоритель; тип вспышки основной камеры

числовое

год выпуска; число sim карт; диагональ экрана; разрешение горизонталь; разрешение вертикаль; разрешение кадр/сек; плотность пикселей; количество цветов экрана млн; количество ядер; частота процессора ггц; объем оперативной памяти гб; объем встроенной памяти гб; максимальный объем карты памяти гб; количество основных камер; количество мегапикселей основной камеры; апертура основной камеры; количество мегапикселей фронтальной камеры; количество фронтальных камер; емкость аккумулятора мА*ч; время работы в режиме разговора; время работы в режиме ожидания; ширина; высота; толщина; вес

логическое

автофокусировка основной камеры; оптическая стабилизация основной камеры; наличие вспышки фронтальной камеры; наличие

конфигурация процессора; датчики; особенности основной камеры; комплектация; биометрическая защита; особенности

Для них также необходимо задать область определения и область значения. Стоит отметить, что для верхних свойств в иерархии будет задаваться только область значения, т.к. такие отношения как «числовое» могут связывать любой «субъект», но только с числом. Окно задание области значения для data properties отличается от аналогичного для object property (рис.10). Для свойств «числовое», «свободное», «текстовое», «логическое» область значения задается с использованием вкладки «Built in datatypes», на которой выбираются типы «xsd:double», «xsd:string», «xsd:boolean».

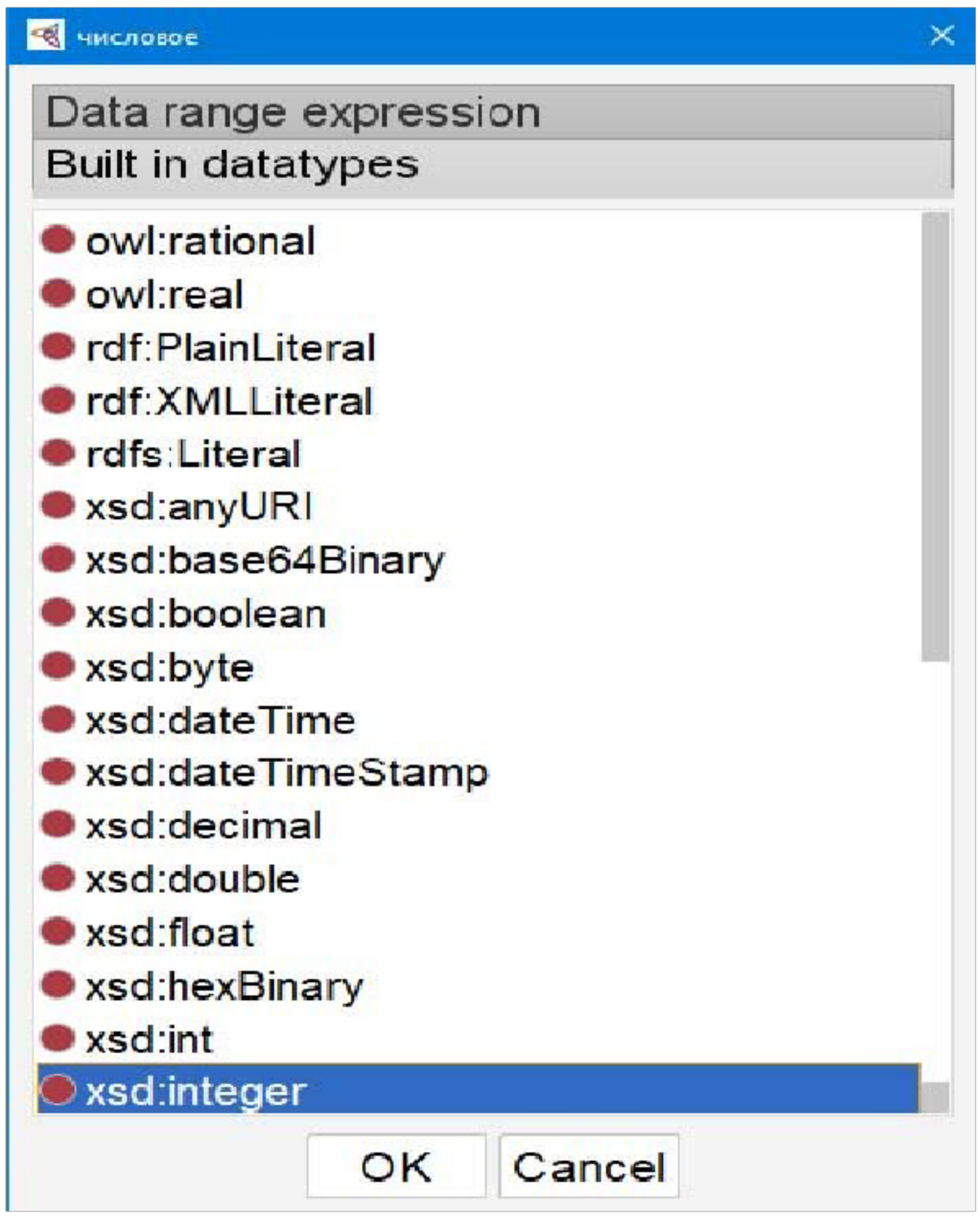



Рис. 10 Окно задания области значения для data property в системе Protégé



Для некоторых свойств необходимо задать дополнительные ограничения. Для этого используется вкладка «Data range expression» и

Manchester Syntax. Разберем некоторые из свойств. Форматы сим карт стандартизированы: Полноразмерные (1FF), Mini-SIM (2FF), Micro-SIM (3FF), Nano-SIM (4FF), Встроенные (E-SIM). Потому можно задать дополнительное ограничение на свойство «формат sim карты» (рис. 11).

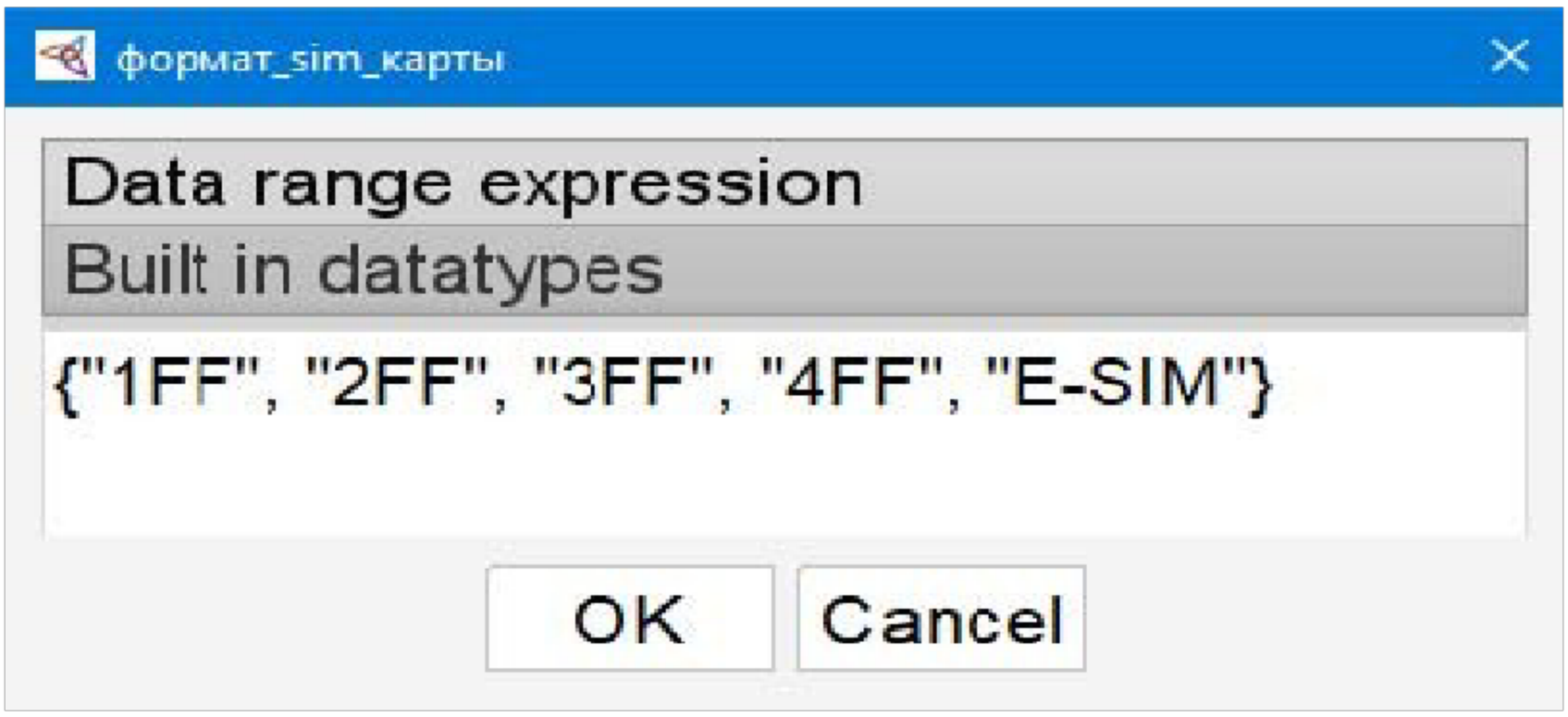



Рис. 11 Задание дополнительных ограничений на свойство «формат sim карты

Рассмотрим, что при этом создается в файле онтологии:

:формат_sim_карты rdf:type owl:DatatypeProperty ;

rdfs:subPropertyOf :текстовое ;

rdfs:range [ rdf:type rdfs:Datatype ; owl:oneOf ("1FF" "2FF" "3FF" "4FF" "E-SIM") ] .

Как видно, для описания области значений создается пустой узел, который является типом данных (экземпляром rdfs:Datatype) и его экземпляры могут принимать значения из заданного списка (за счет использования отношения owl:oneOf). При использовании ограничений на data property пустой узел rdfs:Datatype создается всегда, дальнейшие конструкции зависят от введенного разработчиком. Рассмотрим возможные конструкции для этого поля подробнее:

1 {} – задает список значений, в рамках которых могут быть только скалярные значения (возможно использовать задание типа данных через «^^» и теги языка (@ru))

2 Любой тип данных (xsd:integer, xsd:string и т.д.)

3 Любое скалярное значение

4 [] – ограничение на значение, которые идут после типа значения. Например, xsd:interger[> 0] соответствует целым строго положительным числам. Ограничения прописываются в неограниченном количестве через запятую, например: xsd:integer[> 0 , < 100]. Ограничения, которые могут понадобиться в работах: все знаки сравнения (>, <, >=, <=), проверка строки по регулярному выражению (xsd:string[pattern “[0-9]{4}”]. Необходимо отметить, что регулярное выражение в таком виде проверяет соответствие всей строки, а не ищет подобные вхождения. Т.е. аналогично выражению «^[0-9]{4}$» в обычной ситуации).

5 Логические операторы: or, and. Например, «xsd:integer or xsd:double».

Рассмотрим свойство «год выпуска». Он должен быть целым четырехзначным числом не меньшим 1973 или целым не более чем двузначным числом (в случае, если учитываем возможность внесения сокращенной записи «20», «21», а не делаем преобразование заранее). Пользуясь новыми возможностями, изменим область значений свойства «числовое» на «xsd:integer or xsd:double or xsd:decimal». Далее зададим свойству «год выпуска» ограничение «xsd:integer[>1973] or xsd:integer[>=0 , <= 99]». Также рассмотрим, как это выражение преобразуется в RDF синтаксис (на примере Turtle):

```
:год_выпуска rdf:type owl:DatatypeProperty ;
rdfs:subPropertyOf :числовое ;
rdfs:range [ rdf:type rdfs:Datatype ;
owl:unionOf ( [ rdf:type rdfs:Datatype ; owl:onDatatype xsd:integer ;
owl:withRestrictions ( [ xsd:minInclusive 0 ] [ xsd:maxInclusive
99 ] ) ]
[ rdf:type rdfs:Datatype ; owl:onDatatype xsd:integer ;
owl:withRestrictions ( [ xsd:minExclusive 1973] ) ] ) ] .
```

Приведенное выражение несколько сложнее Manchester Syntax, но позволяет получить понимание как работают выражения в терминах RDF и как они, соответственно, потом обрабатываются.

Создаются 2 пустых узла с отношениями xsd:minInclusive 0, xsd:maxInclusive 99, которые объединяются в RDF список (rdf:List через сокращенное описание «()»). Далее создается новый пустой узел – тип данных, основанный на xsd:integer (owl:onDatatype), которые соединяется отношением owl:withRestrictions с созданным ранее списком. Полностью аналогично создается еще один тип данных, основанный на integer, но с другими ограничениями. После этого создается еще один новый тип данных, которые определяется как объединение (owl:unionOf) созданных ранее. И именно он является областью определения нашего свойства. При этом по факту, созданные типы интересуют нас только с точки зрения возможных значений и могут рассматриваться как множества.

На остальные свойства ограничения могут быть наложены аналогично. Как минимум необходимо заполнить области определения и значения для свойств, связывающих части товара и конкретные характеристики («материал корпуса», «количество ядер» и т.д.). Остальное на данном этапе можно оставить как есть и переходить к экземплярам. Они создаются на вкладке «Individuals». (рис. 12)

На данной вкладке экземпляры создаются по одному. Для этого необходимо нажать на кнопку добавления (рис. 12б) и ввести локальное имя (на основании которого с использование IRI онтологии будет сгенерирован полный IRI), полный IRI или идентификатор с использованием префиксов. Создадим экземпляр рассматриваемого смартфона с идентификатором «SM-G965FZPHSER». После того как экземпляр создан, появляется окно его описания (рис. 13).

Нам необходимо задать экземпляром какого класса является созданная сущность (через отношение rdf:type). Для этого необходимо добавить запись в раздел «Types» (рис 13а), где, использовав вкладку «Class

Hierarchy», выбрать класс «Смартфон».

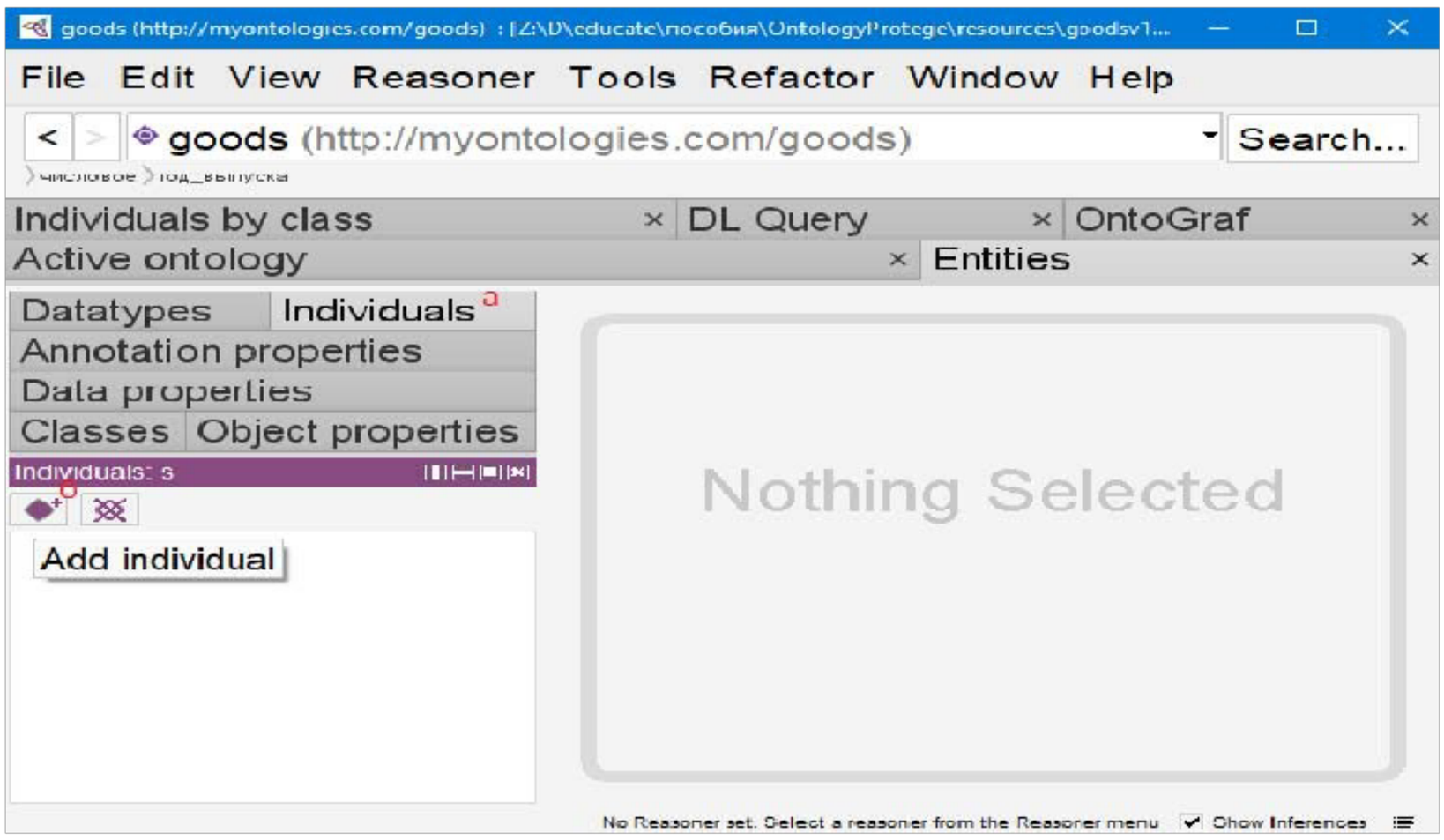



Рис. 12 Вкладка «Individuals» в системе Protégé (а – активная вкладка Individuals, б – кнопка создания нового экземпляра)

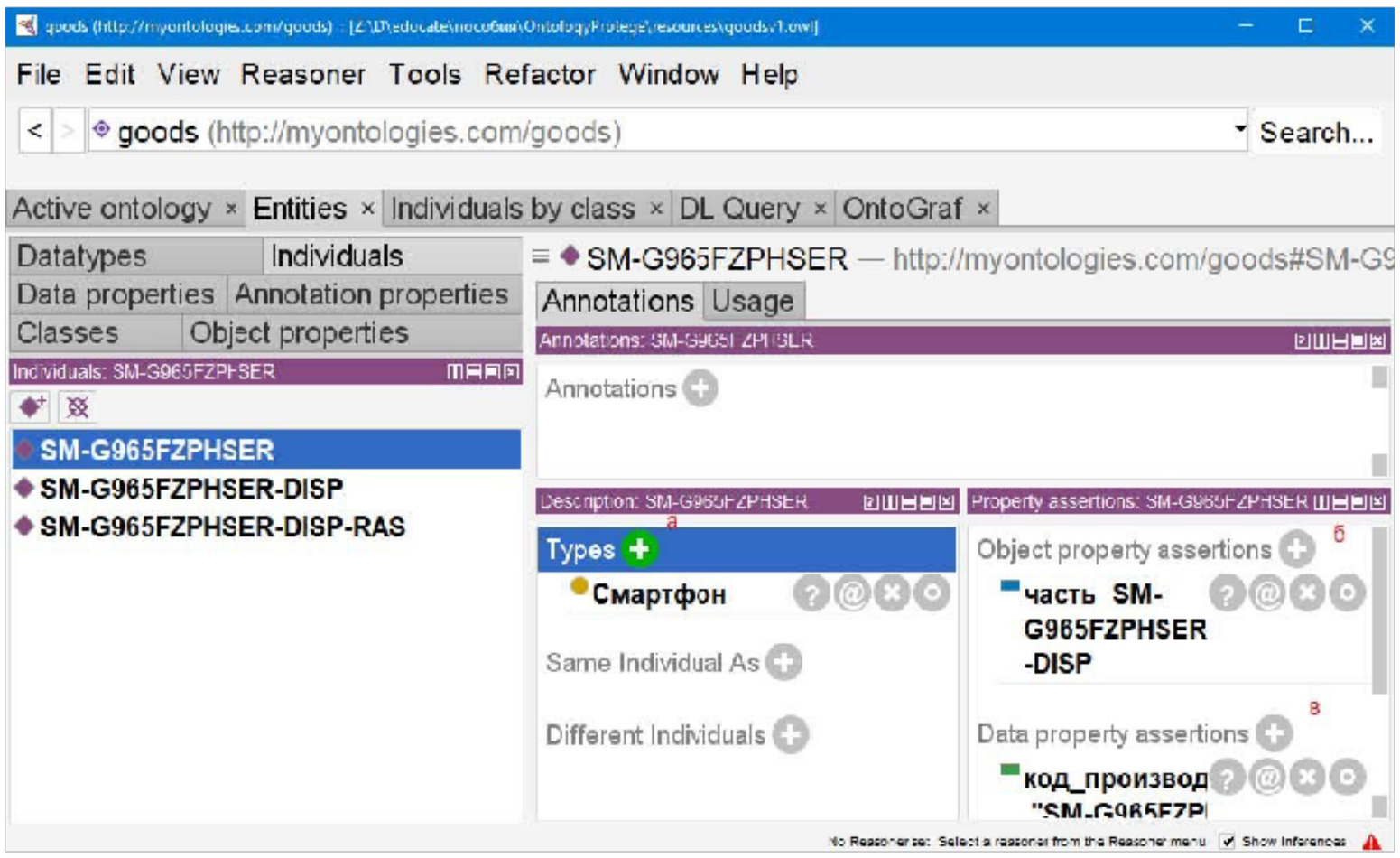



Рис. 13 Окно описания экземпляра в системе Protégé (а – задание отношения rdf:type, б – задание отношений с использование object property, в – задание отношений с использованием data property)



Далее необходимо задать отношения для рассматриваемого смартфона. Первым создадим отношение «часть» с его дисплеем. Для этого по аналогии создадим новый экземпляр класса «Дисплей». После чего для экземпляра

смартфона добавим новую запись в раздел «Object property assertions» (рис. 13б). При добавлении необходимо ввести имя отношения и имя экземпляра, с которым строится
отношение (рис. 14).

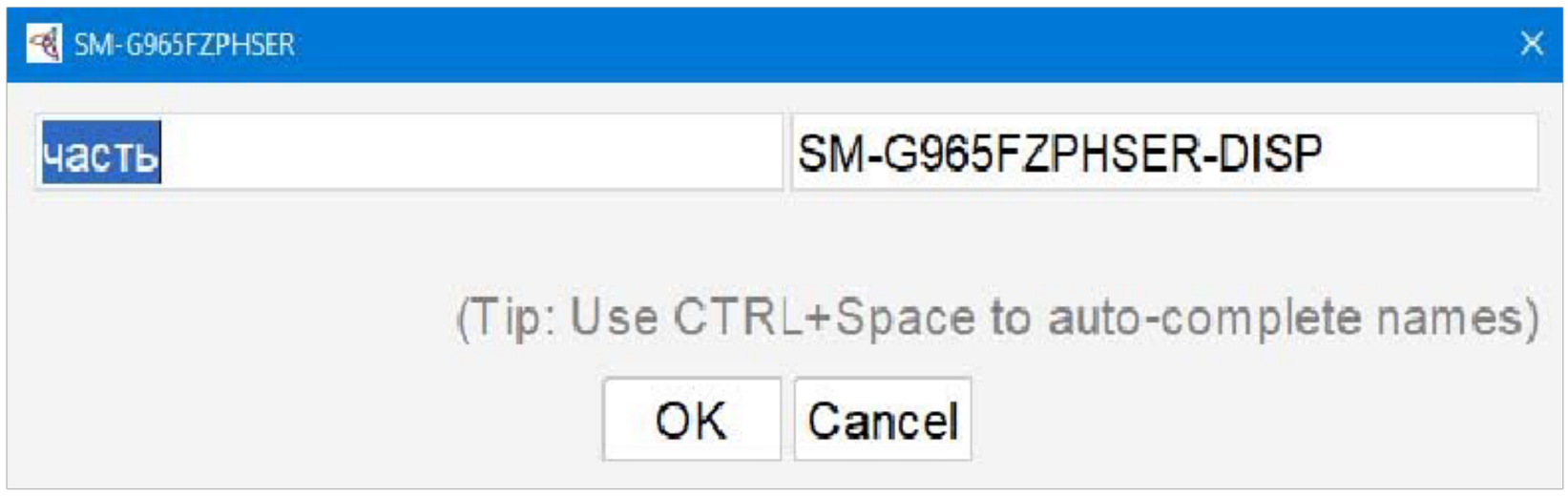



Рис. 14 Окно задания object property для экземпляра

После этого зададим отношение в разделе data properties на примере «код производителя» (рис. 15).

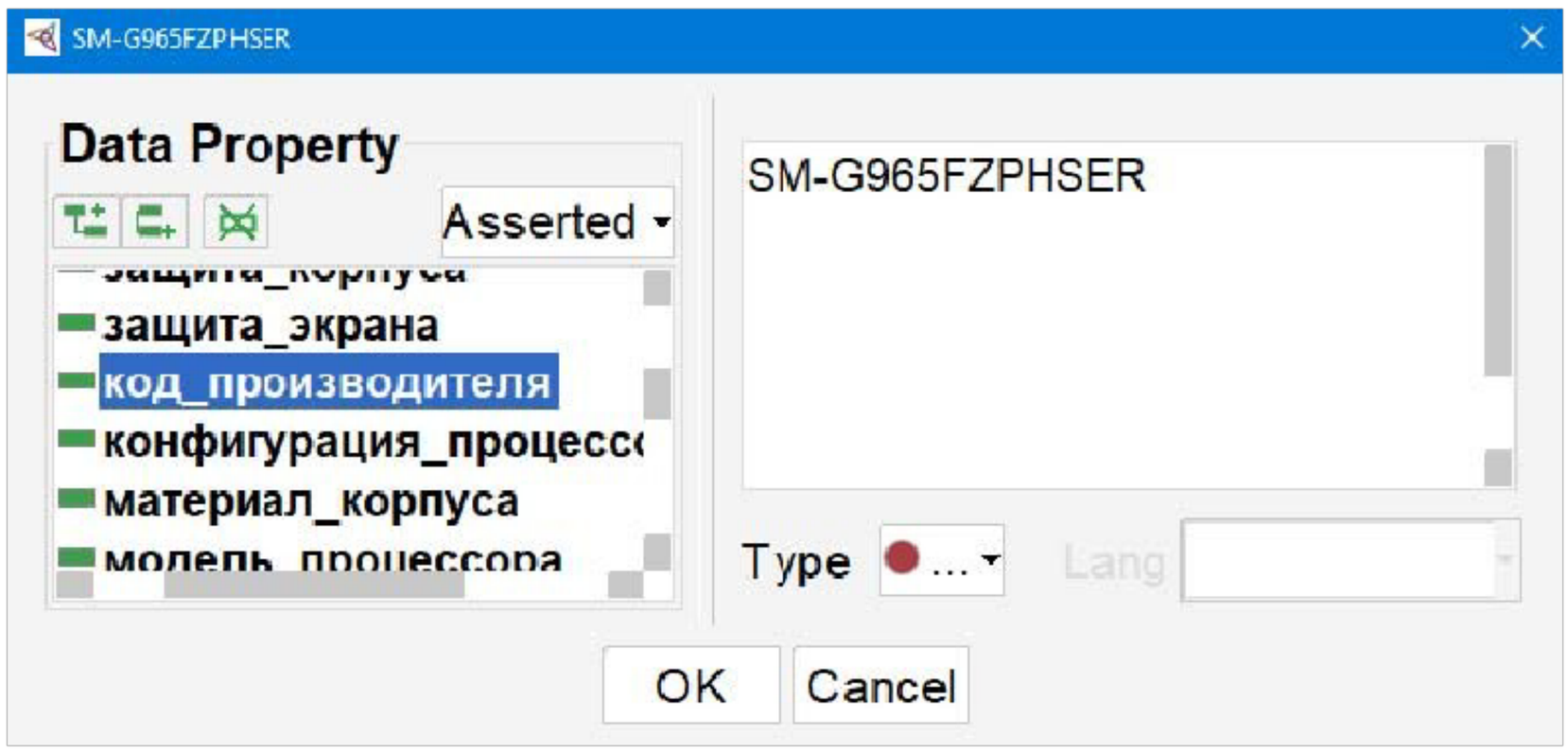



Рис. 15 Окно задания data property для экземпляра

Для этого в появившемся окне необходимо выбрать само свойство, ввести его значение и выбрать тип.

Для определенных типов также активируется поле выбора языка.

По аналогии создаем экземпляр «Разрешение» и задаем дисплею разрешение.

Рассмотрим текстовое представление созданных сущностей:

:SM-G965FZPHSER rdf:type owl:NamedIndividual , :Смартфон ;

:часть :SM-G965FZPHSER-DISP ;

код_производителя "SM-G965FZPHSER"^^xsd:string .

:SM-G965FZPHSER-DISP rdf:type owl:NamedIndividual , :Дисплей ;

:разрешение :SM-G965FZPHSER-DISP-RAS .

:SM-G965FZPHSER-DISP-RAS rdf:type owl:NamedIndividual , :Разрешение ;

:разрешение_вертикаль 1920 ;

:разрешение_горизонталь 1080 .

Новая синтаксическая конструкция «,» означает, что после нее будет записан только «объект» из тройки, а «субъект – предикат» будут взяты из предыдущей. «owl:NamedIndividual» указывает, что сущность является экземпляром и не пустым узлом. Однако, на данном этапе, при удалении этого отношения, в системе Protégé не будет видна разница. Дальнейшее описание достаточно большого количества экземпляров и их отношений проще сделать напрямую в файле онтологии с использованием любого текстового редактора. При этом можно было бы заменить некоторые экземпляры на пустые узлы:

:SM-G965FZPHSER-DISP rdf:type

owl:NamedIndividual , :Дисплей ; :разрешение

[rdf:type :Разрешение ; :разрешение_вертикаль

1920 ; :разрешение_горизонталь 1080] .

Система будет понимать такое описание, однако в интерфейсе не будет указываться сам экземпляр класса «Разрешение» и, как следствие, не будет возможности редактирования его описания. При этом все возможности обработки, в том числе вывод в рамках системы Protégé остаются возможными. Тем не менее, для большей наглядности при открытии онтологии в системе, далее будет приведен текст экземпляров и их отношений без использования таких пустых узлов.

:SM-G965FZPHSER rdf:type owl:NamedIndividual , :Смартфон ;

:часть :SM-G965FZPHSER-DISP;

:часть :Exynos_9810;

:часть :SM-G965FZPHSER-CAM-BACK;

:часть :SM-G965FZPHSER-CAM-FRONT;

:часть :SM-G965FZPHSER-BAT;

:имя_производителя "SM-G965FZPHSER";

:цвет_заявленный_производителем "ультрафиолет";

:сеть_2g "GSM 1900", "GSM 1800", "GSM 900";

:сеть_3g "WCDMA 900", "WCDMA 2100";
:частота_lte "LTE 1800 (B3)", "LTE 2100 (B1)", "LTE 2300 (B40)",
"LTE 2600 (B7)", "LTE 1900 (B39)", "LTE 2000 (B34)";
:формат_sim_карты "4FF";
:степень_защиты "IP68";
:операционная_система "Android 8.0 Oreo";
:год_выпуска 18;
:число_sim_карт 2;
:объем_оперативной_памяти_гб 6;
:объем_встроенной_памяти_гб 256;
:максимальный_объем_карты_памяти_гб 400;
:материал_корпуса "металл", "стекло";
:защита_корпуса "защита от пыли и влаги";
:ширина 73.8;
:высота 158.1;
:толщина 8.5;
:вес 189;
:датчики "гироскоп, датчик Холла, акселерометр, барометр, компас, датчик приближения, датчик освещения, датчик сердечного ритма";
:комплектация "комплект амбушюр, комплект переходников, документация, сетевой адаптер, скрепка для извлечения слота SIM-карты, наушники";
:биометрическая_защита "сканер лица, сканер радужной оболочки глаза, сканер отпечатков пальцев";
:особенности "ANT+, технология Dolby Atmos, камера Bixby, камераперевод чик, селфимоджи, стереодинамики от AKG, несъемный аккумулятор" .

:SM-G965FZPHSER-DISP rdf:type owl:NamedIndividual , :Дисплей ;
:разрешение :SM-G965FZPHSER-DISP-RAS;
:технология_изготовления_экрана "Super AMOLED";
:защита_экрана "Corning Gorilla Glass";
:диагональ_экрана 6.2;
:плотность_пикселей 529;
:количество_цветов_экрана_млн 16 . :SM-G965FZPHSER-DISP-RAS rdf:type owl:NamedIndividual ;
:разрешение_вертикаль 1920 ;
:разрешение_горизонталь 1080 .
:Exynos_9810 rdf:type owl:NamedIndividual , :Процессор ;
:производитель_процессора "Samsung" ;
:модель_процессора "Exynos 9810" ;
:конфигурация_процессора "4x Custom 2.9 ГГц, 4x Cortex-A55 1.9 ГГц" ; :количество_ядер_процессора 8 ; :частота_процессора_ггц 2.9, 1.9 ;
:графический_ускоритель "Mali-G72 MP18" .
:SM-G965FZPHSER-CAM-BACK rdf:type owl:NamedIndividual , :Камера ;

:количество_основных_камер 2 ;
:количество_мегапикселей_основной_камеры 12, 12 ;
:апертура_основной_камеры 1.5, 2.4 ;
:тип_вспышки_основной_камеры "светодиодная" ;
:разрешение :SM-G965FZPHSER-DISP-CAM-BACK-RAS-1, :SM G965FZPHSER-DISP-CAM-BACK-RAS-2, :SM-G965FZPHSER-DISP-CAM-BACKRAS-3;
:автофокусировка_основной_камеры
1 ; :оптическая_стабилизация_основной_камеры 1 ;
:особенности_основной_камеры "сверхбыстрый датчик изображения, двойная апертура объектива" .
:SM-G965FZPHSER-DISP-CAM-BACK-RAS-1 rdf:type owl:NamedIndividual ;
:разрешение_вертикаль 3840 ;
:разрешение_горизонталь 2160 ;
#полный IRI необходим из-за спец символа "/"
<http://myontologies.com/goods#разрешение_кадр/сек> 60 .
:SM-G965FZPHSER-DISP-CAM-BACK-RAS-2 rdf:type owl:NamedIndividual ;
:разрешение_вертикаль 1280 ;
:разрешение_горизонталь 720 ;
#полный IRI необходим из-за спец символа "/"
<http://myontologies.com/goods#разрешение_кадр/сек> 960 .
:SM-G965FZPHSER-DISP-CAM-BACK-RAS-3 rdf:type owl:NamedIndividual ;
:разрешение_вертикаль 1920 ;
:разрешение_горизонталь 1080 ;
#полный IRI необходим из-за спец символа "/"
<http://myontologies.com/goods#разрешение_кадр/сек> 240 .
:SM-G965FZPHSER-CAM-FRONT rdf:type
owl:NamedIndividual , :Камера ; :количество_мегапикселей_фронтальной_камеры 8; :количество_фронтальных_камер 1 ; :разрешение :SM-G965FZPHSER-DISP-CAM-FRONT-RAS-
1 ; :наличие_вспышки_фронтальной_камеры 0 .
:SM-G965FZPHSER-DISP-CAM-FRONT-RAS-1 rdf:type owl:NamedIndividual , :Разрешение ;
:разрешение_вертикаль 2560 ;
:разрешение_горизонталь 1440 ;
#полный IRI необходим из-за спец символа "/"
<http://myontologies.com/goods#разрешение_кадр/сек> 30 .
:SM-G965FZPHSER-BAT rdf:type owl:NamedIndividual , :Аккумулятор ;
#полный IRI необходим из-за спец символа "*"
<http://myontologies.com/goods#емкость_аккумулятора_мА*ч> 3500 ; :время_работы_в_режиме_разговора 25 ;
:наличие_беспроводной_зарядки 1 .

Теперь основные данные внесены и можно проверить онтологию на непро
тиворечивость. Для этого необходимо выбрать механизм вывода HermiT
Reasoner
[5] (рис. 16б) и запустить его (рис. 16в).

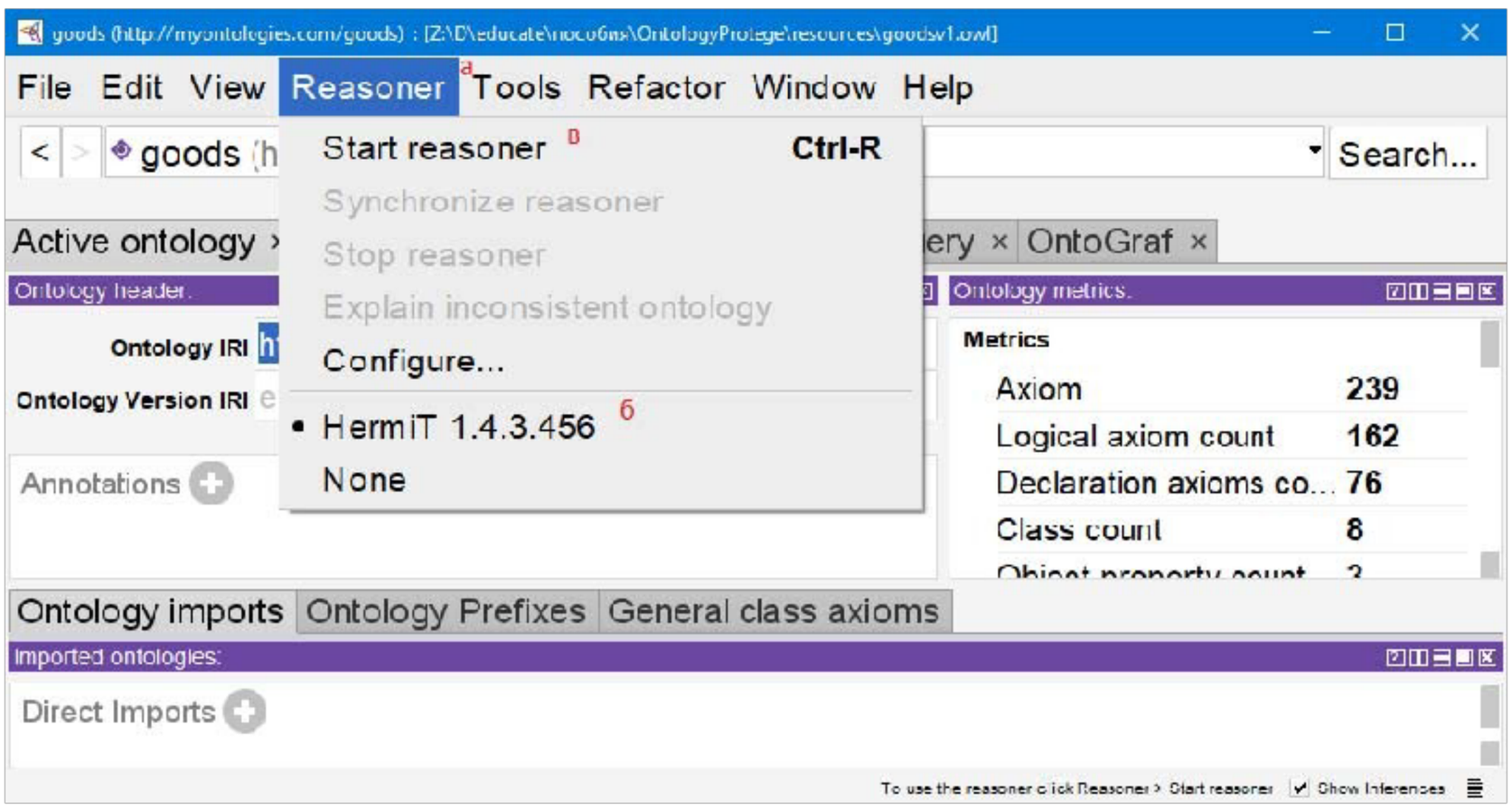



Рис. 16 Запуск механизма вывода в системе Protégé (а – пункт меню с действиями вывода, б – выбранный механизм HermiT, в – кнопка запуска вывода).

Индикацией того, что вывод запущен в данном случае будет то, что кнопка «Start reasoner» станет неактивной, а «Stop reasoner» наоборот активной. Если в процессе вывода были найдены противоречия, то система выведет сообщения об ошибках с их объяснением (рис. 17а) или выведет сообщение об ошибке запуска вывода (рис. 17в). При выполнении всех приведенных выше действий и рекомендаций ошибок быть не должно.

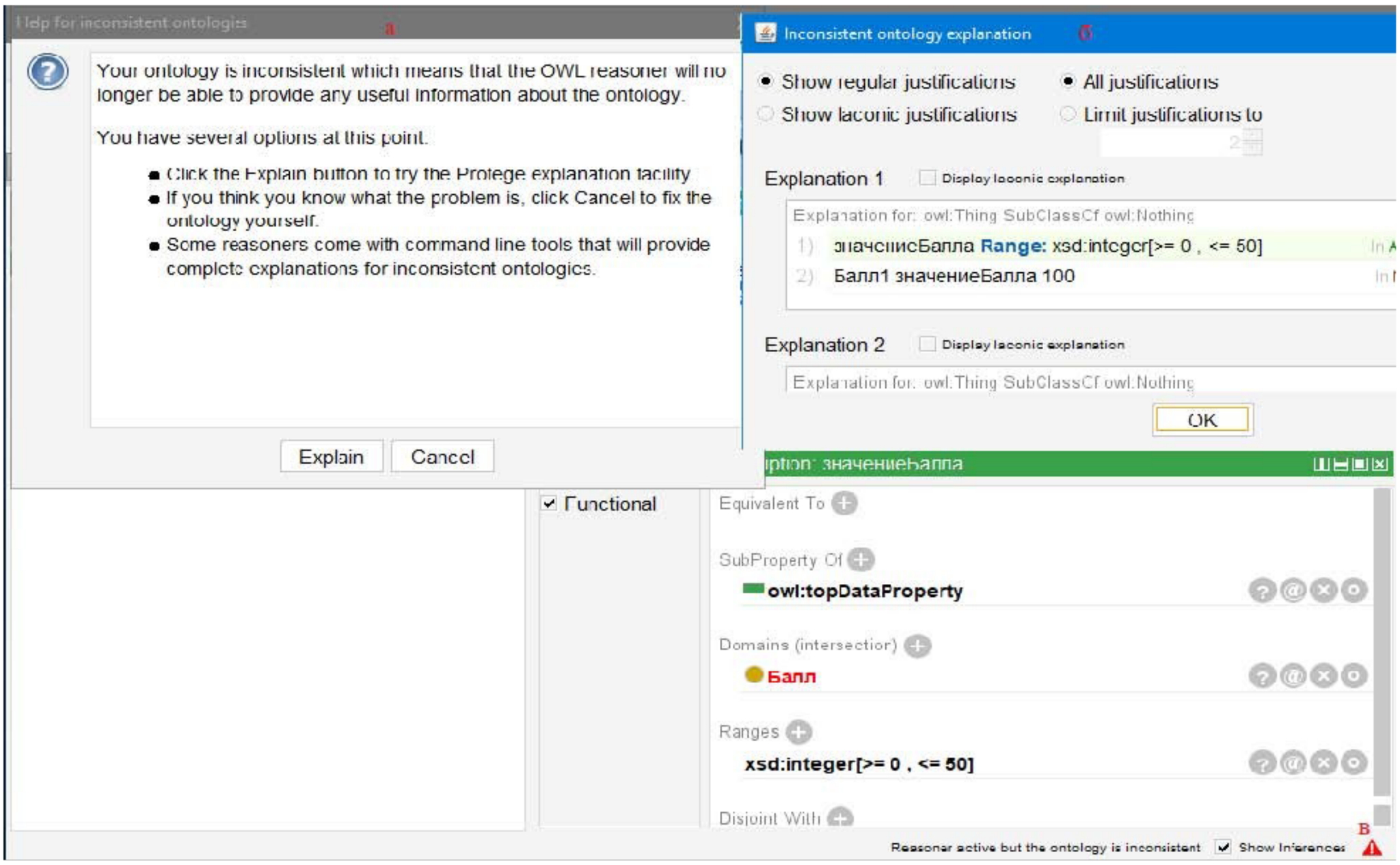



Рис. 17 Пример ошибки при запуске механизма рассуждений (а – сообщение о проблемах в онтологии с предложением объяснения; б – окно объяснений; в – индикатор ошибок в логах системы)

Также, если внимательно изучить приведенный выше фрагмент с экземплярами онтологии, то мы намеренно не задали тип для таких сущностей как «G965FZPHSER-DISP-CAM-BACK-RAS-2». Если при этом читатель выполнял все рекомендации и заполнил область определения для таких отношений как «разрешение_горизонталь», то можно увидеть, что тип был выведен автоматически (рис. 18). Выведенные (Inferred) понятия отмечаются в системе Protégé желтым цветом.

Для удобства дальнейшей обработки дополним онтологию новым object property: «является_частью». Это свойство является обратным к «часть», что необходимо отметить в его описании (рис. 19а). Прописывать его к созданным сущностям нет необходимости, т.к. оно будет выведено (рис. 19б).

В процессе этой работы было создана первая версия онтологии, декларативно описывающей предметную область «товар смартфон». В следующих работах в онтологию будут добавлены новые аксиомы, выражения и правила для вывода отношений и проверки непротиворечивости БЗ, а также написаны запросы для выбора информации из нее по определенным критериям.

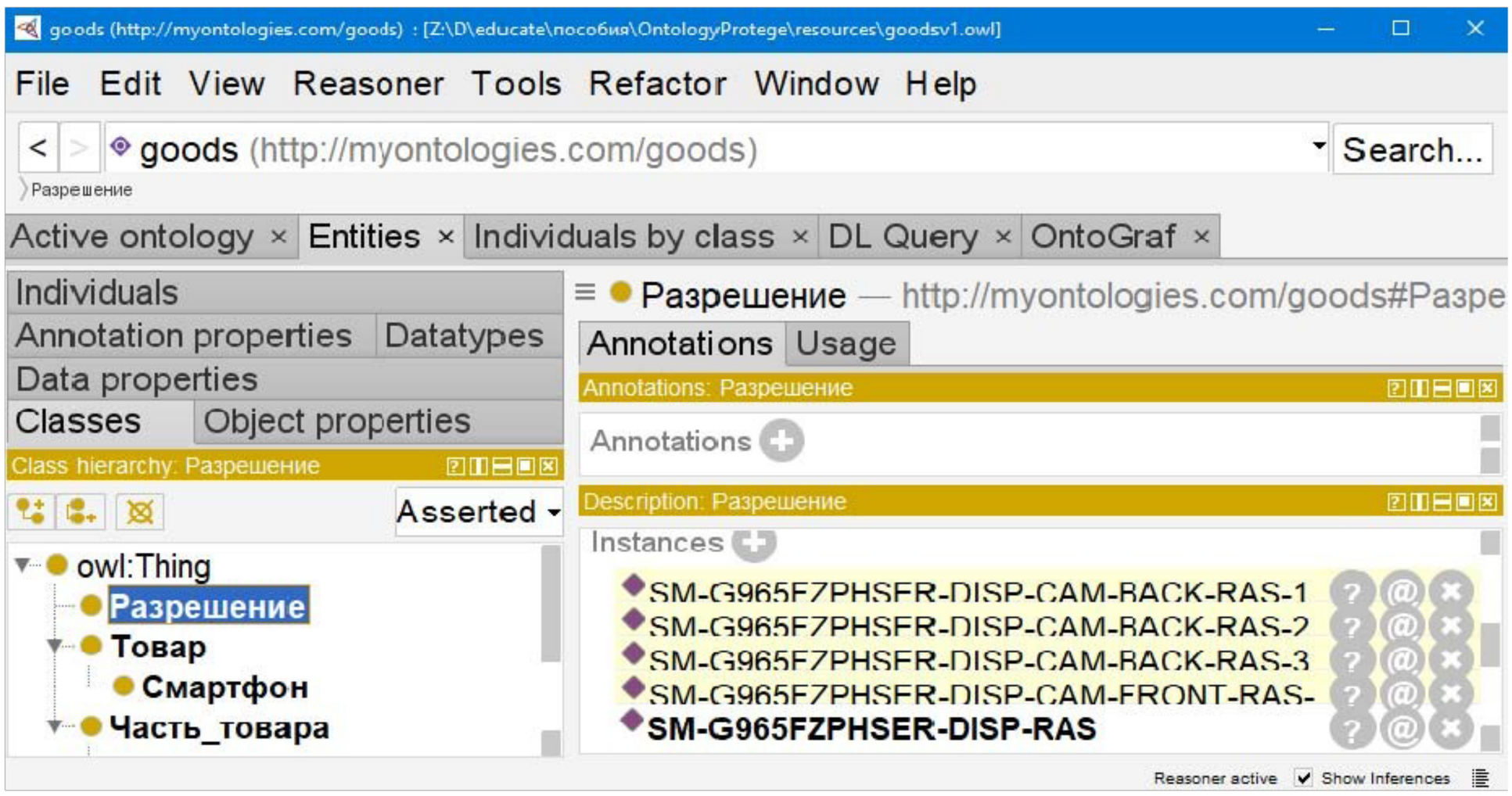



Рис. 18 Выведенные экземпляры у класса «Разрешение» в системе Protégé


Рис. 19 Демонстрация вывода в соответствии с описанием inverseOf

**Задания**

1 Дополнить созданную базу знаний минимум 10 экземплярами других смартфонов

2 По аналогии, дополнить базу знаний для категорий «ноутбук» и «холодильник»

## Создание defined classes в системе PROTÉGÉ

В этой работе будут созданы классы, отношения «rdf:type» с которыми не будут указываться в явном виде, а будут строиться автоматически в процессе вывода. Такие классы в системе Protégé называются «Defined Classes» («обычные» классы называются «Primitive Classes»). Для примера будут реализованы следующие классы: «Новый смартфон», «Камерофон», «Игровой смартфон», «Смартфон для фильмов», «Смартфон для походов». Это позволит образовать в онтологии группы экземпляров в соответствии с заранее заданными критериями, которые будут доступны для дальнейшей обработки.

Первоначально сформируем критерии, по которым экземпляры будут относится к заявленным классам:

1 Новый смартфон. Выпущен после 18 года.

2 Камерофон. Минимум 2 основные камеры. Минимум 12 мегапикселей. Есть оптическая стабилизация. Съемка видео минимум Ultra HD 4K 60кадр./сек.

3 Игровой смартфон. Процессор Snapdragon или Exynos. Минимум 6 ядер. Частота работы процессора минимум 2.9ГГц. Объем оперативной памяти минимум 6гб. Графический ускоритель Mali или Adreno.

4 Смартфон для фильмов. Игровой смартфон. Соотношение сторон 21:9. Разрешение экрана от 4K.

5 Смартфон для походов. Емкость аккумулятора от 6000 мА*ч. Степень защиты выше чем IP68. Защита экрана «Gorilla Glass». Защита корпуса

«укрепленный корпус». Материал корпуса включает металл.

Далее отладим запросы, в соответствии с которым будут формироваться экземпляры обозначенных классов. Такие запросы формируется с использованием Manchester Syntax и называются в системе Protégé «DL Query». Для отладки таких запросов выделена отдельная вкладка (рис. 20).

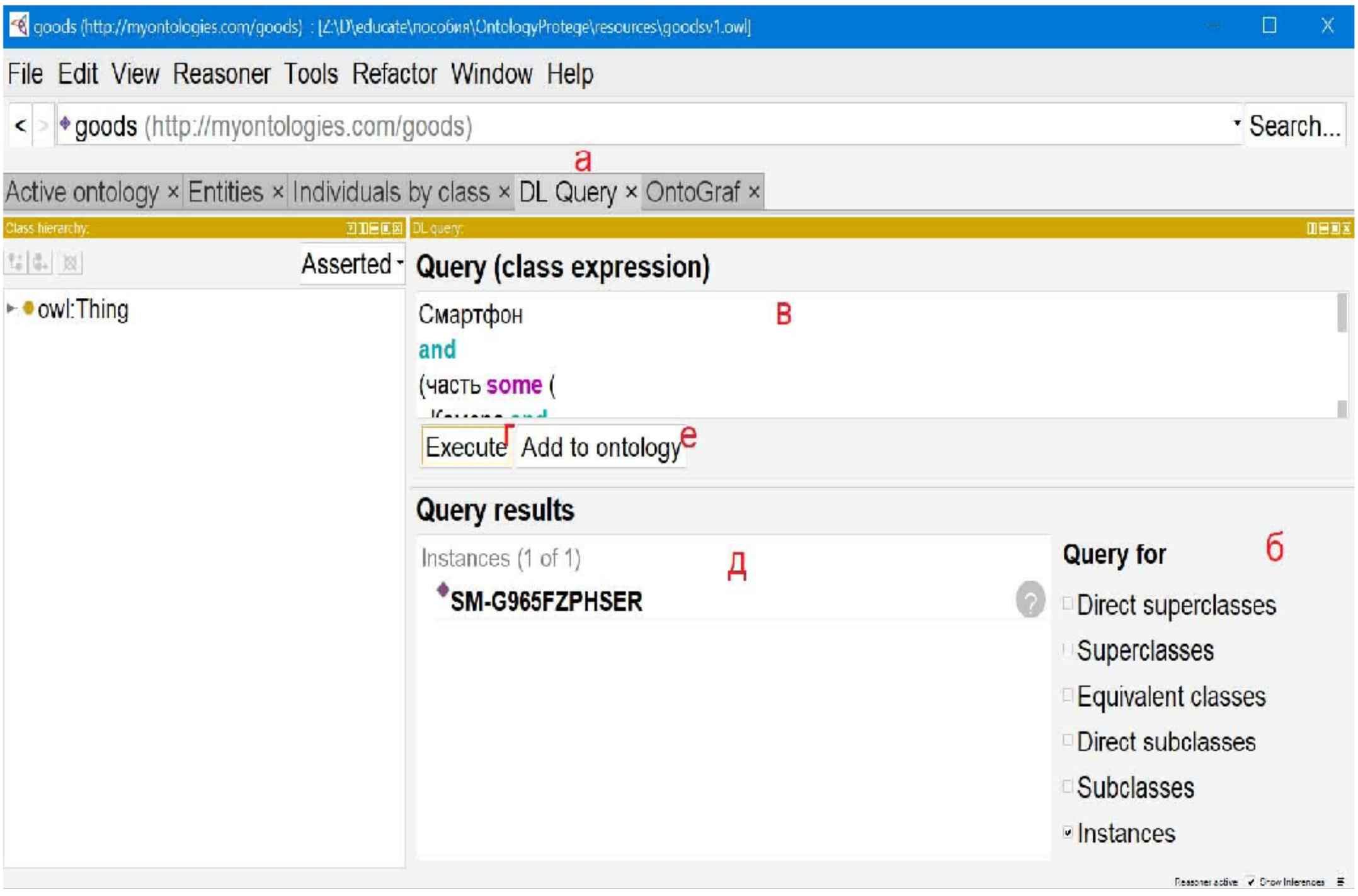



Рис. 20 Вкладка «DLQuery» в системе Protégé (а – панель вкладок с активной вкладкой «DLQuery»; б – панель выбора типа запрашиваемых сущностей; в – поле ввода запроса; г – кнопка выполнения запроса; д – результаты запроса; е – кнопка добавления запроса в онтологию)

Для освоения интерфейса выберем все экземпляры класса «Смартфон». Для этого необходимо в поле ввода запроса (рис. 20в) ввести запрос «Смартфон», выбрать ожидаемый тип результата – экземпляры (рис. 20б), запустить запрос (рис. 20г). После этого можно увидеть результаты (рис. 20д).

Перед написанием запросов необходимо определиться с возможными для использования конструкциями (объяснения будут приводится применительно к типу выбираемых сущностей – экземпляр):

1. Локальное имя класса. Позволяет выбрать экземпляры данного класса. Как явно заданные, так и выведенные. Например: «Смартфон».

2. <локальное_имя_свойства> some <выражение>. Позволяет выбрать все экземпляры, для которых задано как минимум одно отношение <локальное_имя_свойства> с сущностью, подходящей под <выражение>. В качестве выражения могут выступать: вложенный «DL Query»; любое вариант ограничений, рассмотренный ранее при задании области значения data property. Например: «год_выпуска some xsd:integer»

3. <локальное_имя_свойства> only <выражение>. Все экземпляры, для которых отсутствуют заданные отношения или они все удовлетворяют выражению. Например: «год_выпуска only xsd:integer[ >= 18].

4. <локальное_имя_свойства> exactly <количество>. Все экземпляры, для которых задано указанное количество указанных отношений. Например: «год_выпуска exactly 1»

5. <локальное_имя_свойства> exactly <количество> <выражение>. Все экземпляры, для которых задано указанное количество указанных отношений, удовлетворяющих выражению. При этом может быть сколько угодно таких же отношений, не удовлетворяющих выражению. Например: «год_выпуска exactly 1 xsd:integer[ >= 18]»

6. <локальное_имя_свойства> min/max <количество> [<выражение>]. Аналогично exactly, но должно быть не менее/не более, чем указанное количество указанных отношений.

7. <локальное_имя_свойства> value <литерал>. Все экземпляры, которые связаны указанным отношением с указанным скалярным значением.

8. <выражение1> and/or <выражение2>. Пересечение/объединение множеств выбранных экземпляров выражения1 и выражения2. Например: «Дисплей or Процессор»

9. not <выражение>. Можно рассматривать как операция дополнения над множеством экземпляров, выбранных в рамках выражения. Например: «not Смартфон»

Определившись с множеством возможных конструкций, составим запрос для первого класса «Новый смартфон». У экземпляров такого класса не должно быть отношений «год выпуска», связывающих их с числом меньше 18 (для упрощения в первом варианте запроса рассмотрим год как двузначное число). Но, при этом должно быть хоть одно такое отношение (если таких отношений нет, то недостаточно информации для принятия решения). Также экземпляр должен быть типа «Смартфон». Соответственно, необходимый нам запрос должен описываться как пересечение 3 множеств: «Смартфон and (год_выпуска min 1) and (год_выпуска only xsd:integer[>= 18])».

Выполнив этот запрос на вкладке «DL Query», можно увидеть, что результатов не найдено. Вывод основывается на теории дескриптивных логик. Однако, для понимания вывода и дальнейших действий, рассмотрим причины пустого результата в упрощенном варианте.

В онтологии относительно года выпуска есть как минимум утверждения «:SM-G965FZPHSER rdf:type Смартфон; :год_выпуска 18». Для того, чтобы экземпляр был выбран в результатах запроса, необходимо, чтобы он относился к каждому из трех множеств. Учитывая приведенные утверждения, можно сделать вывод, что «:SM-G965FZPHSER» относится к классу «Смартфон» и «год_выпуска min 1». Но, нельзя сделать никакие выводы относительно «год_выпуска only xsd:integer[>= 18]». У нас информация о том, что существует такое x(«:SMG965FZPHSER :год_выпуска x»), которое удовлетворяет ограничениям (x >= 18), но из этого нельзя сделать вывод, что для любого x это ограничение всегда выполняется. При этом [illegible] (Смартфон and (год_выпуска min 1) and (год_выпуска only xsd:integer[>= 18]))» рассматриваемый экземпляр также не

войдет. Т.е. для «год_выпуска only xsd:integer[>= 18]» применительно к данного экземпляру не был сделан вывод ни о ложности, ни о истинности высказывания. В этом (предположение об открытости мира) состоит еще одна существенная разница с базами данных.

Одним из вариантов решения сложившейся проблемы является внесение в онтологию новых сведений. Так, если бы для рассматриваемого экземпляра выполнялось ограничение «:год_выпуска exactly 1» (для рассматриваемого экземпляра существует ровно одно отношение «год выпуска»), то с учетом существования подходящего под критерии отношения уже можно было бы сделать вывод о том, что существуют только такие отношения. Эти сведения могут быть добавлены в разделе «Types» экземпляра с использованием вкладки «Class expression editor» (рис. 21).

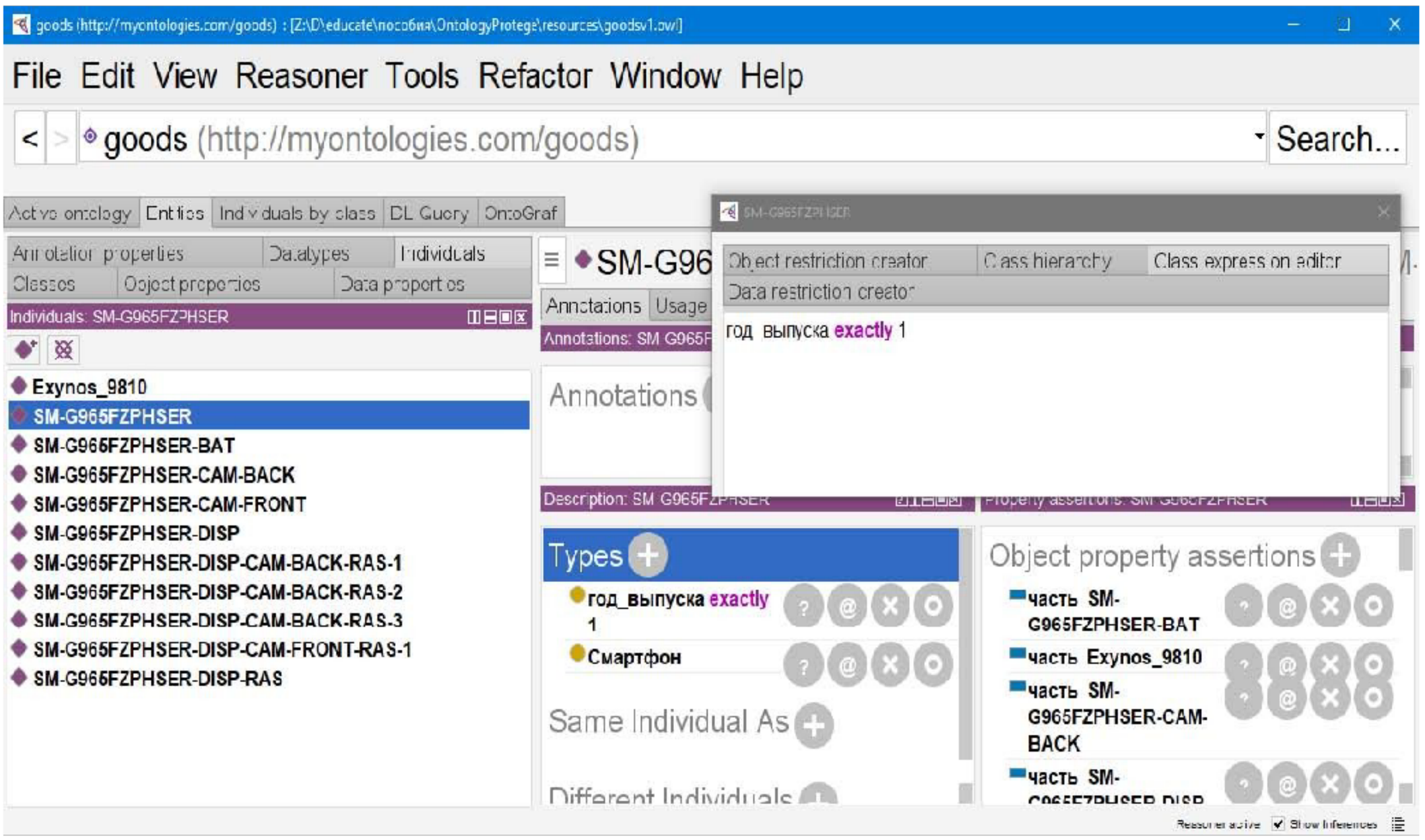



Рис. 21 Использование вкладки «Class expression editor» для добавления информации о экземпляре



После отладки запроса необходимо добавить класс. Это можно сделать с использованием кнопки «Add to ontology» на вкладке «DL Query» (рис. 20е)

или создав класс и заполнив поле «Equivalent To» в его описании (рис. 22).

Рассмотрим, как данный класс описывается в тексте онтологии.

:Новый_смартфон rdf:type owl:Class ;
owl:equivalentClass [ owl:intersectionOf ( :Смартфон
[ rdf:type owl:Restriction ;
owl:onProperty :год_выпуска ;
owl:allValuesFrom [ rdf:type rdfs:Datatype ; owl:onDatatype
xsd:integer ; owl:withRestrictions ( [ xsd:minInclusive 18 ] ) ] ]
[ rdf:type owl:Restriction ; owl:onProperty :год_выпуска ;
owl:minCardinality "1"^^xsd:nonNegativeInteger ] ); rdf:type
owl:Class ]

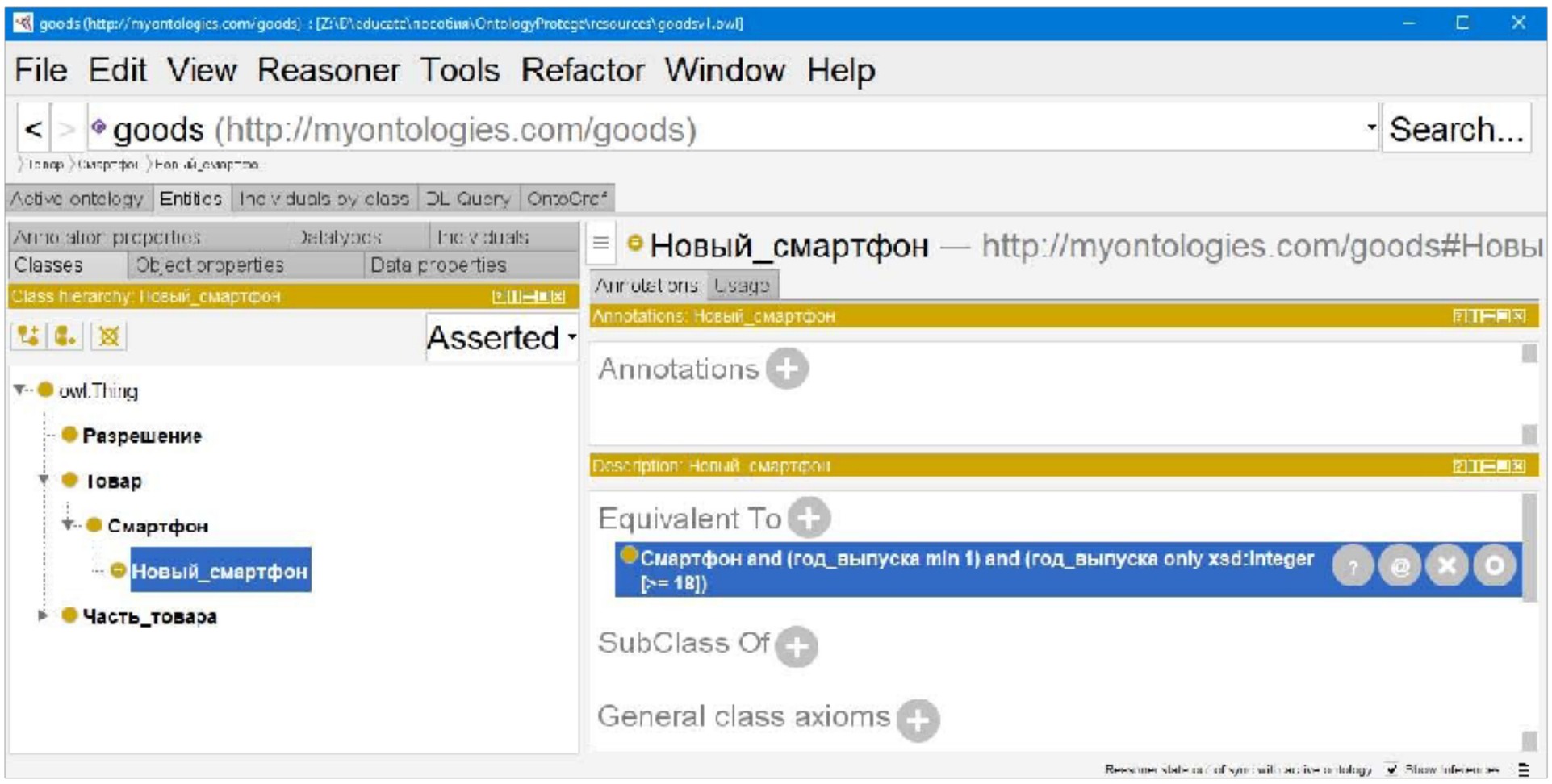



Рис. 22 Пример заполнения поля «Equivalent To» в описании класса

Как видно, используется отношение «owl:equivalentTo», которое позволяет сделать вывод, что если экземпляры относятся к «объекту» этого отношения, то относятся и к «субъекту». А в качестве «объекта» выступает пустой узел, являющийся пересечением 3 других классов, как и планировалось изначально.



Дополним написанный запрос для работы с годом, заданным в полном формате:

Смартфон and (год_выпуска min 1) and (год_выпуска only (xsd:integer[>= 18,

<= 25] or xsd:integer[>= 2018]))

Также составим запрос для класса «Камерофон»:

Смартфон

and

(часть some ( Камера and (количество_основных_камер some xsd:integer[ >= 2

])

and (количество_мегапикселей_основной_камеры some xsd:integer[ >= 12 ])

and

(разрешение some (Разрешение and

((разрешение_горизонталь some xsd:integer[ >=

3840 ]) or (разрешение_вертикаль some

xsd:integer[ >= 3840 ])) and

(разрешение_кадр/сек some xsd:integer[ >= 60 ])

)) )

)

**Задания**

1 Составить запросы для оставшихся групп: «Смартфон для фильмов», «Игровой смартфон», «Смартфон для походов». Создать defined class для каждой.

2 Сформировать минимум 5 групп (на основании имен подборок на ресурсе источнике) для категорий «Ноутбуки» и «Холодильники». Для каждой из групп сформировать критерии выбора экземпляров и создать defined class.

## Написание запросов на языке SPARQL



В предыдущей работе было произведено отнесение экземпляров к группам по заданным критериям. Для этого использовались defined classes и язык DL Query. Данные инструменты сильно расширяют

возможности рассуждений в рамках онтологий. Однако, DL Query применяется именно в рамках рассуждений и имеет ряд недостатков, если использовать его только для получения доступа к явно заданным или уже выведенным сущностям онтологии: медленное выполнение (т.к. используется логический вывод); необходимость учитывать поведение при выводе (например, добавления в онтологию «exactly» в прошлой работы); невозможность описания некоторых условий (например, выбор отношений, а не экземпляров; выбор ряда характеристик смартфона). Поэтому для таких задач используется другой язык – SPARQL [8]. Данный язык позволяет выбирать и добавлять сущности в онтологию, формировать новый граф RDF. В данном работе будут рассмотрены только возможности выбора. Язык синтаксически схож с SQL (и не только синтаксически. В частности, он также основан на предположении о закрытости мира, в отличие от DL Query), применяемым для аналогичных задач в базах данных. Поэтому (и потому что SQL должен быть уже знаком обучаемым) с ним будут проводиться аналогии.

Для работы с SPARQL в системе Protégé необходимо активировать вкладку «Window -> Tabs -> SPARQL Query» (рис. 23). Необходимо понимать, что SPARQL и механизм рассуждений в Protégé работают независимо и не влияют на результаты работы друг друга. Это означает, что нет возможности выбрать экземпляры класса «Новый смартфон», которые являются выведенными. При этом это возможно в рамках, например, библиотеки Apache Jena для языка Java.

Первоначально выберем первые 2 смартфона и характеристики их камер (количество камера; мегапиксели; апертура) и рассмотрим на примере этого запроса основные синтаксические конструкции.

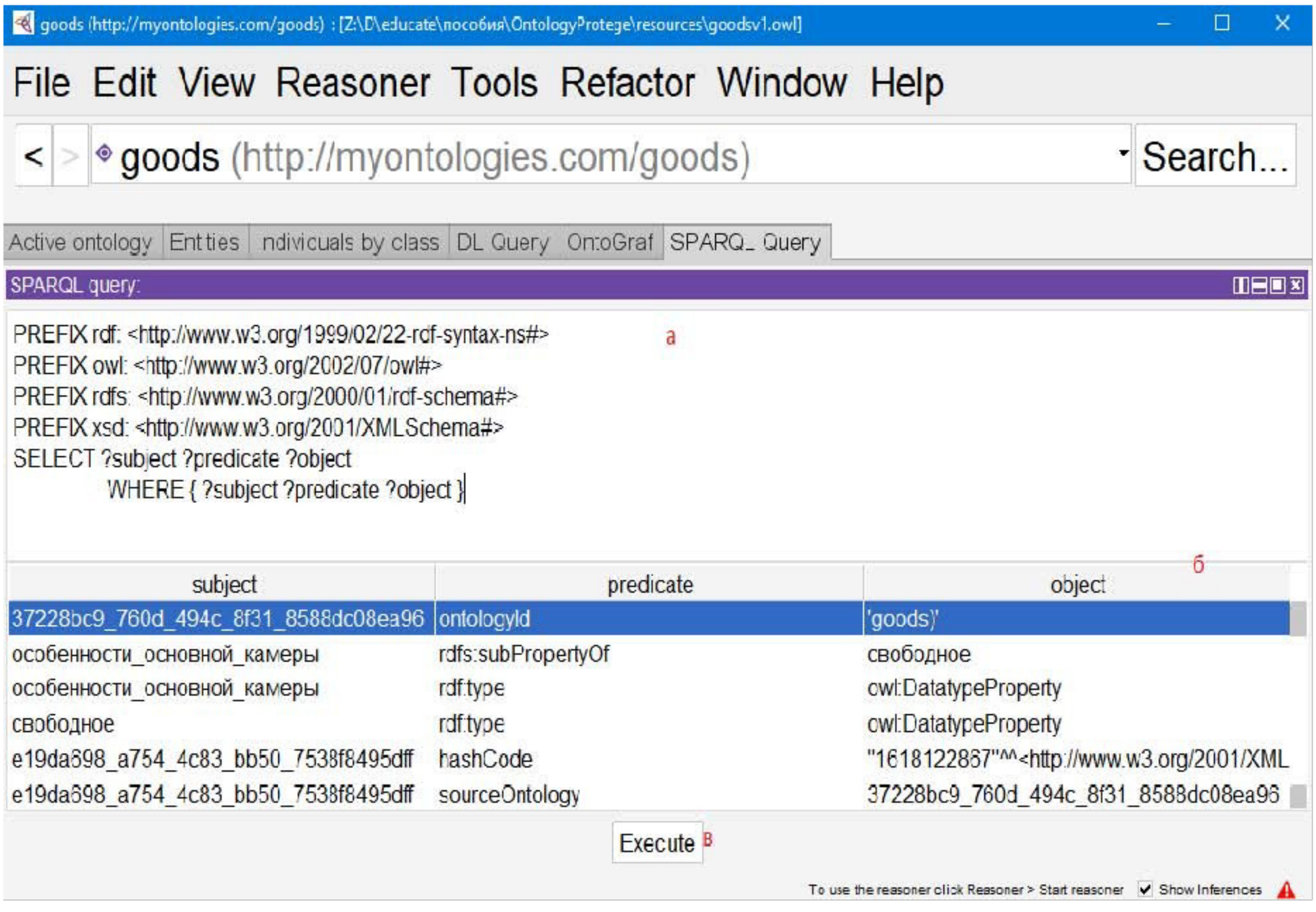



Рис. 23 Вкладка «SPARQL Query» в системе Protégé (а – поле ввода запроса; б – результаты запроса; в – кнопка выполнения запроса)

#Сначала приводится блок стандартных префиксов для того, чтобы не записывать громоздкие идентификаторы в рамках основной части запроса

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX owl: <http://www.w3.org/2002/07/owl#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

#Также необходимо добавить префикс для созданной онтологии (ее идентификатор берется с вкладки «Active Ontology»)

PREFIX : <http://myontologies.com/goods#>

#После префиксов идет конструкция SELECT{переменные}, которая описывает что будет выбрано в качестве результата запроса

#Все переменные начинаются с «?» и перечисляются через пробел

SELECT ?смартфон ?камера_число ?камера_мп ?камера_ап

#Далее идет конструкция WHERE { {тройки} }. Тройки записываются аналогично синтаксису Turtle, что позволяет также использовать разделители «.», «;», «,».

```
WHERE {
#Вместо отдельных сущностей могут ставится переменные, начинающиеся с «?». На них могут накладываться условия, они могут выбираться в рамках списка полей в «SELECT».
    ?смартфон :часть ?камера .
    ?камера rdf:type :Камера ;
      :количество_основных_камер ?камера_число ;
      :количество_мегапикселей_основной_камеры ?камера_мп ;
      :апертура_основной_камеры ?камера_ап }
#Далее идет конструкция ORDER BY, позволяющая задать сортировку результатов
    ORDER BY DESC(?камера_ап)
#Далее идет конструкция LIMIT, OFFSET позволяющая задать количество пропускаемых записей и количество выбираемых
    LIMIT 2 OFFSET 0
```

Как видно, структура запроса схожа с языком SQL (в том числе доступны конструкции GROUP BY, HAVING), однако содержимое блоков различается. Основная часть – WHERE, разберем подробнее ее вложенные конструкции, которые будут использоваться в рамках работы:

1. Тройка «субъект предикат объект». На месте каждого из элементов тройки может стоять конкретный IRI или литерал или переменная. Позволяет выбрать все тройки онтологии, подходящие под заданную (заданная тройка используется как маска). При выборе: на месте переменных в заданной тройке может стоять любая сущность в выбираемой тройке; литералы должны совпадать по значению, типу, языку; IRI на соответствующих местах должны совпадать.



2. Разделители «.» между тройками можно интерпретировать как декартового

произведение, но с учетом равенства соответствующих переменных. Например, «?смартфон :код_производителя ?код» позволяет выбрать все тройки, в которых присутствует отношение «код производителя». Если в онтологии 2 смартфона и только у одного из них есть это отношение, то будет выбрана 1 тройка.

«?смартфон :часть ?часть» позволяет выбрать все тройки, в которых присутствует отношение «часть». Если в онтологии 2 смартфона и у каждого из них по 5 частей, то будет выбрано 10 троек.

Если упрощенно рассмотреть порядок выбора в рамках «?смартфон :код_производителя ?код . ?смартфон :часть ?часть», то будет сначала выбрана 1 запись по первой тройке, 10 записей по второй. Далее будет сформировано 10 записей как все варианты совмещений данных записей. После чего будут удалены 5 записей, где в рамках одноименных переменных «смартфон» не совпадают значения.

3. В рамках троек также могут использоваться пустые узлы, описываемые аналогично синтаксису Turtle с использованием «[]».

4. OPTIONAL { {вложенная часть where} }. Позволяет задать вложенный запрос, который будет использован только при наличии результатов. Например, «?смартфон :код_производителя ?код . ?смартфон :часть ?часть» выберет 10 записей, если в онтологии будет 2 смартфона и 5 частей у каждого. Но, если будет 2 смартфона и 0 частей у каждого, то никаких результатов выбрано не будет. При этом «?смартфон :код_производителя ?код OPTIONAL { ?смартфон :часть ?часть }» выберет также 10 записей в первом случае, но 2 записи во втором (часть в рамках OPTIONAL учитываться не будет).

5. FILTER({условие}). Позволяет оставить в рамках выбранных записей только те, которые подходят под условие. Например, «?смартфон :год_выпуска ?год FILTER(?год > 18)» позволяет выбрать только те результаты, где год выпуска позднее 18 (стоит учитывать, что если у смартфона 2 таких отношения и у одного значение больше 18, а у другого

меньше, то сам смартфон все равно будет выбран, т.к. одна из записей с ним будет подходить под условия). Т.к. FILTER не является тройкой, то после него нет необходимости использовать разделители. Таких конструкций в запросе может быть множество. В рамках условий FILTER могут использоваться операции: &&, ||, <, >, <=, >=, =, !=. Также могут использоваться встроенные функции, например, regex, isLiteral (список всех функций здесь приводиться не будет и может быть найден в документации).

6. FILTER EXISTS { { вложенная часть where} }, FILTER NOT EXISTS { { вложенная часть where} }. Специальные виды FILTER, которые оставляют записи, для которых вложенный запрос вернул/не вернул результаты.

Зная основные конструкции, составим следующий запрос: выбор информации (идентификатор, код производителя, конфигурация процессора, емкость аккумулятора, количество мегапикселей основной камеры, диагональ экрана) о всех смартфонах не старее 2018 года с «Super AMOLED» экраном.

Необходимо учитывать, что часть информации может отсутствовать, но при этом результат все равно должен быть представлен, значит необходимо использовать множество конструкций OPTIONAL. Условия на год и экран являются обязательными и сущности, не имеющие этой информации, не должны выбираться. Условие на экран накладывается через механизм троек. Условие на год должно учитывать, что могло быть переиздание смартфона и несколько таких отношений. Значит необходимо или работать с минимальным годом или использовать конструкцию FILTER NOT EXISTS.

```
SELECT ?смартфон ?код ?конфигурация ?диагональ ?емкость ?мп
WHERE {
  #с помощью пустого узла определяем условие, что выбираемый
  идентификатор должен быть "Смартфон" или его подкласс
  ?смартфон rdf:type [ rdfs:subClassOf* :Смартфон ] .
```

#условие, что связь "год выпуска" должна существовать. Реализовано вложенным условием, чтобы в случае нескольких годов не выбралось несколько записей

FILTER EXISTS {?смартфон :год_выпуска ?год}

#условие на "новизну" года выпуска

FILTER NOT EXISTS {?смартфон :год_выпуска ?год2 FILTER(?год2 < 18 || (?год2 > 1000 && ?год2 < 2018))}

#считаем, что дисплей один. Для нескольких выберется информация только о подходящих

?смартфон :часть ?дисплей . ?дисплей rdf:type :Дисплей .

#условие на технологию изготовления экрана

?дисплей :технология_изготовления_экрана ? технология FILTER(regex(?технология, "super amoled", "i"))

#дополнительная выбираемая информация (при наличии)

OPTIONAL {?смартфон :код_производителя ?код}

OPTIONAL {?смартфон :часть ?проц . ?проц rdf:type :Процессор . ? проц :конфигурация_процессора ?конфигурация}

OPTIONAL {?дисплей :диагональ_экрана ?диагональ}

OPTIONAL {?смартфон :часть ?аккум . ?аккум rdf:type :Аккумулятор . ?аккум <http://myontologies.com/goods#емкость_аккумулятора_мА*ч> ? емкость } OPTIONAL {?смартфон :часть ?камера . ?камера rdf:type :Камера . ?камера :количество_мегапикселей_основной_камеры ? мп } }

## Задания

Сформировать запросы:

1 выбор всех смартфонов и информации о них (аналогичной описанной в п.12.1) с аккумулятором не менее 500 мА*ч

2 выбор всех вариантов блоков камер смартфонов с их информацией

(апертура, количество камер, производитель, мегапиксели, разрешение видеосъемки)

3 выбор всех вариантов модулей экранов с их информацией (диагональ, разрешение, плотность пикселей, технология изготовления экрана)

4 выбор первых 10 ноутбуков с их характеристиками: конфигурация процессора, тактовая частота, диагональ экрана, разрешение экрана, размер оперативной памяти, размер носителя информации, тип носителя информации, список разъемов

## Написание правил с использованием языка SWRL

В рамках последней работы необходимо сформировать в онтологии правила, которые позволят, учитывая особенности предметной области, формировать выведенные отношения. Для примера рассмотрим правило, которое будет отмечать сведения об аккумуляторе смартфона как сомнительные, если заявленное время работы в режиме разговора превышает «емкость аккумулятора / 250».

SWRL [9, 10] позволяет составлять правила, являющиеся импликацией между предпосылкой и следствием. После запуска механизма рассуждений онтология будет изменяться в соответствии с этими правилами. Предпосылками и следствиями являются выражения на основе предикатов (в том числе созданных в рамках онтологий). Рассмотрим синтаксис на примерах:

1 Смартфон(?x). Означает ?x rdf:type :Смартфон (?x аналогичен переменным в языке SPARQL)

2 часть(?x, ?p) – означает ?x :часть ?p. Таким образом можно записать предикат на основании любого созданного data или object property.

3 Смартфон(?x) ^ часть(?x, ?p). операция «and» записывается как «^».

4 Смартфон(?x) ^ часть(?x, ?p) -> является_частью(?p, ?x). Операция импликации записывается как «->». Данное правило определяет аксиому о

том, что если ?x является смартфоном и содержит часть ?p, то также можно сказать, что ?p является частью ?x. Подобное правило генерируется автоматически в соответствии с полем «inverseOf», которое мы указывали в прошлых работах.

Из встроенных предикатов [9] будут необходимы: swrlb:lessThanOrEqual(?x, ?y) (?x меньше или равно ?y); swrlb:lessThan(?x, ?y) (?x строго меньше ?y); swrlb:greaterThanOrEqual(?x, ?y) (?x больше или равно ?y; swrlb:greaterThan(?x, ?y) (?x строго больше ?y); swrlb:divide(?x, ?y, ?z) (в ?x результат операции ?y/?z).

Для отладки условных частей правил может использоваться язык SQWRL [11]. В рамках него будет использоваться только предикат sqwrl:select(?x, ?y, ?z, ). Он позволяет отобразить пользователю все перечисленные сущности, для которых активировались предпосылки. Таким образом можно преобразовать правило в аналог запроса и проверить, что оно применяется для запланированных сущностей.

Сформируем SQWRL правило по заданным в работе критериям: «Аккумулятор(?a) ^ емкость_аккумулятор_мА_ч(?a, ?v) ^ время_работы_в_режиме_разговора(?a, ?t) ^ swrlb:divide(?tr, ?v, 250) ^ swrlb:greaterThan(?t, ?tr) -> sqwrl:select(?a, ?t, ?tr)» Именно такие записи о аккумуляторах будут отмечаться сомнительными. Чтобы проверить корректность условий данного правила следует рассмотреть для каких аккумуляторов это правило будет срабатывать (выбрать идентификатор аккумулятора, заявленное время работы и вычисленное время работы).

Для доступа к SQWRL необходимо активировать вкладку «Window -> Tabs > SQWRL Tab» (рис. 24). Далее необходимо создать правило (рис. 24б) и ввести его тело (рис. 25в). Далее необходимо выделить правило и нажать кнопку запуска (рис. 24а, рис. 24в). После этого результат работы можно увидеть в таблице, содержащей имени правила (рис. 25г).

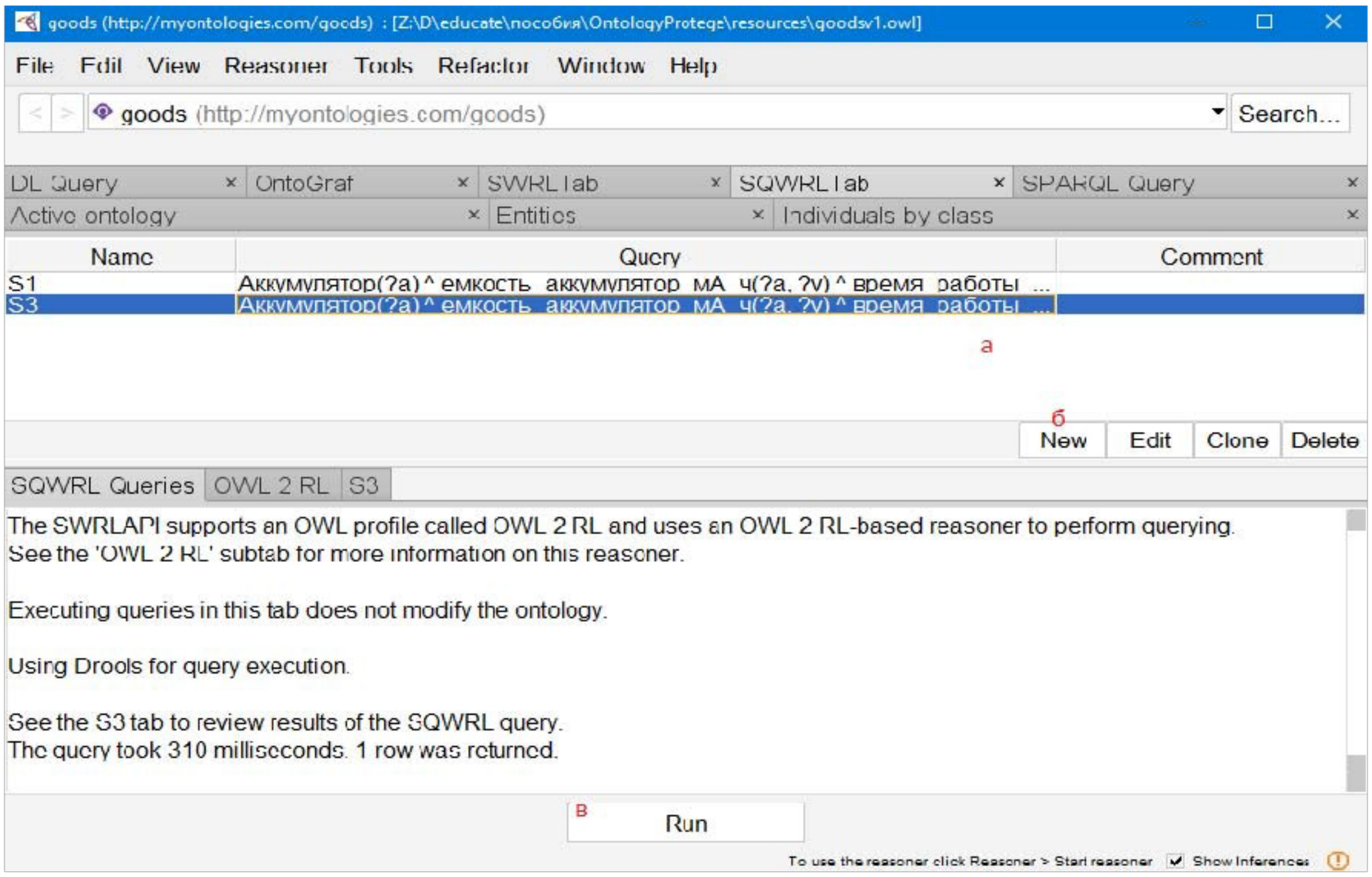



Рис. 24 Вкладка SQWRL в системе Protégé (а – список созданных правил; б – кнопки создания нового правила и редактирования выделенного; в – кнопка запуска выбранного правила)

Убедившись в корректности условной части правила, можно перенести его в список SWRL правил (Window -> Tabs -> SWRL Tab) и изменить его заключение. На данной вкладке уже отображаются все SQWRL правила, однако они отмечены как неактивные. Можно или изменить правило и отметить его активным или создать новое. Условная часть правила остается прежней, а заключением будет являться «Сомнительно(?a)» (также необходимо создать соответствующий класс).

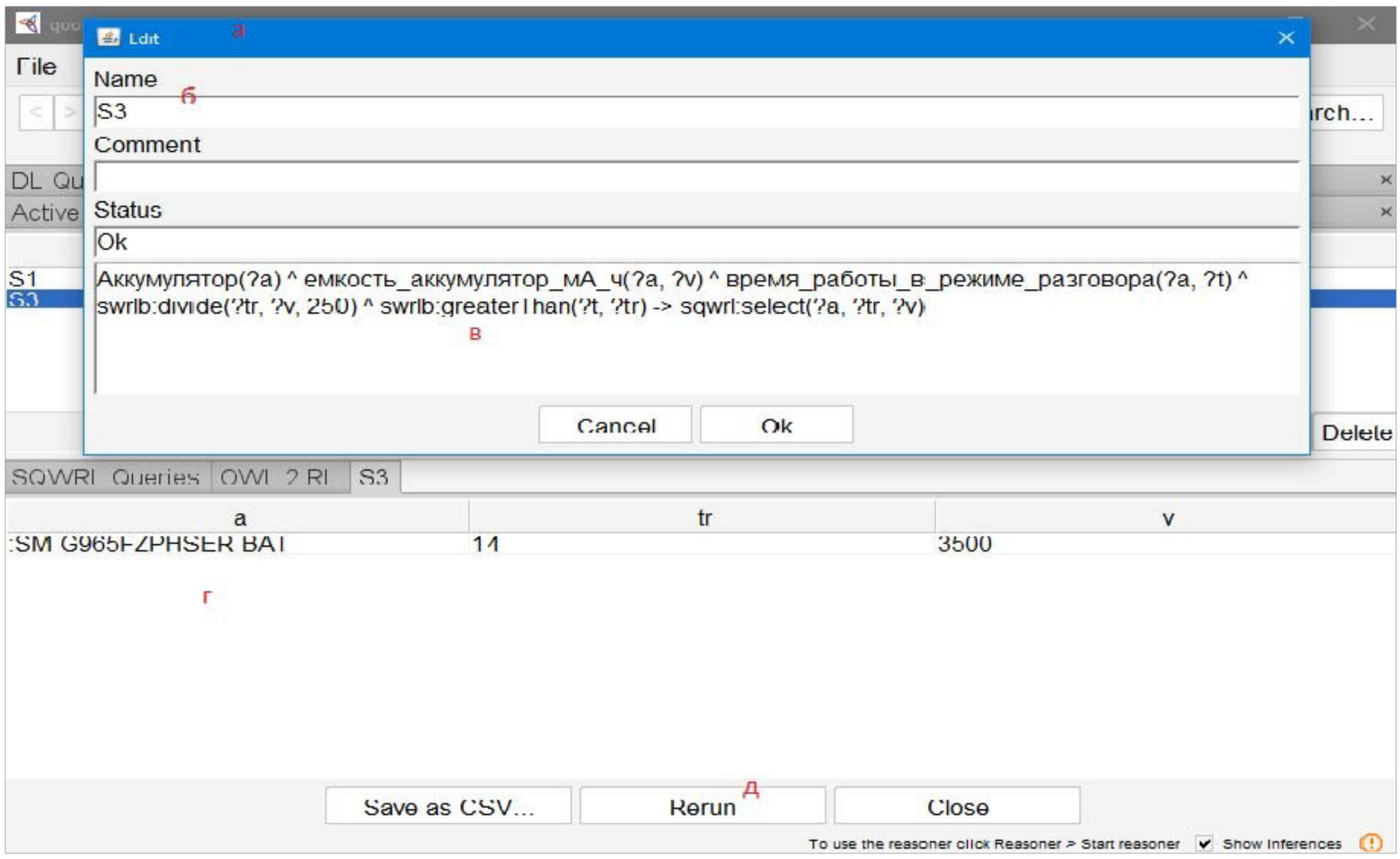



Рис. 25 вкладка SQWRL в системе Protégé (а – окно редактирования правила; б – имя правила; в – тело правила; г – результаты выполнения правила; д – кнопка перезапуска правила)

Теперь необходимо запустить правило, предварительно загрузив понятия из онтологии (рис. 26б, рис. 26в).

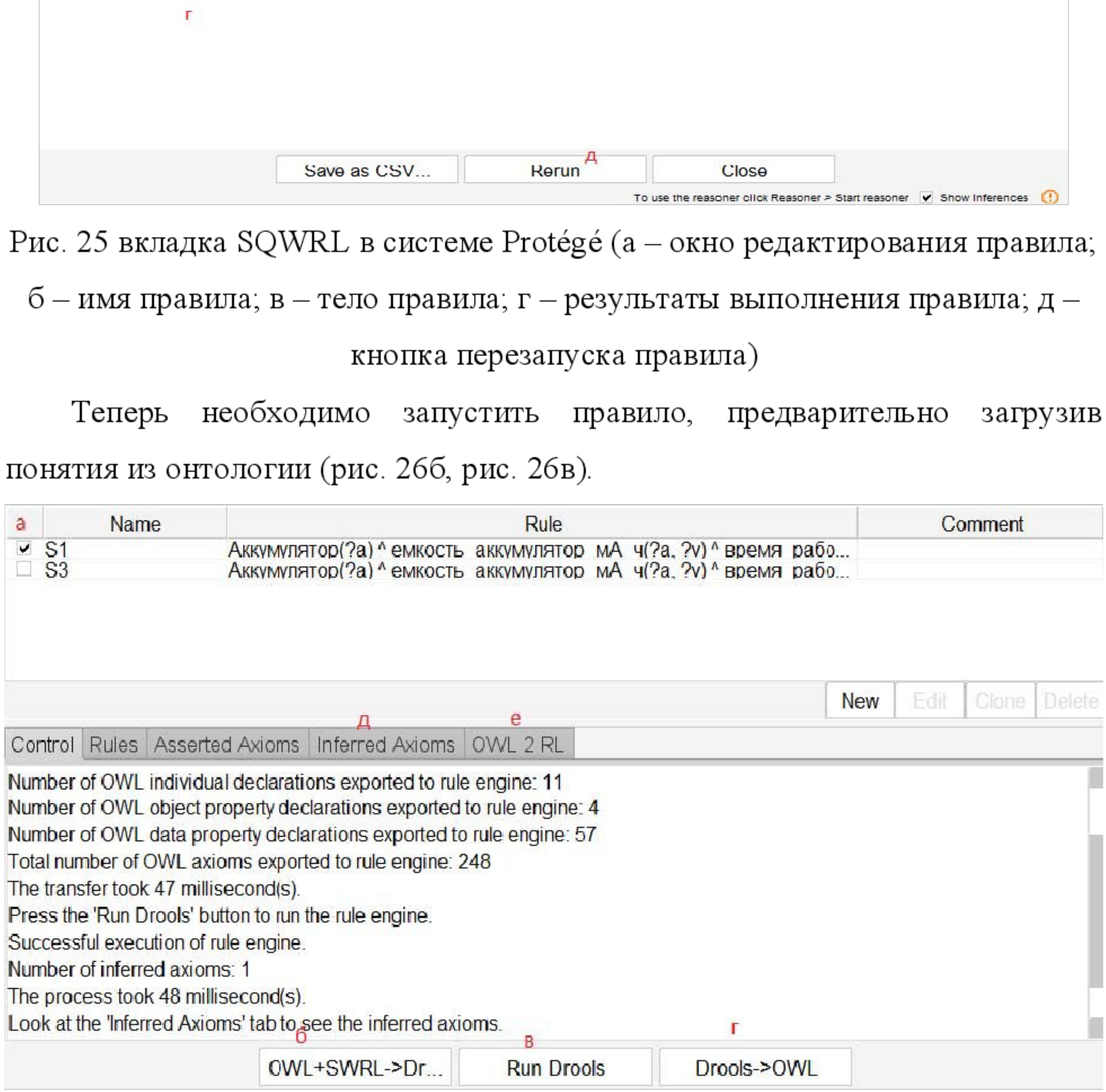



Рис. 26 Вкладка SWRL в системе Protégé (а – активность правила; б – кнопка загрузки понятий и аксиом из онтологии; в – запуск вывода на основе правил; г – внесение выведенных понятий в онтологию; д – список

выведенных понятий; e – настройки вывода)

После этого можно ознакомиться с выведенными аксиомами (рис. 26.д). В случае настроек по умолчанию их будет больше 100. Это связано с тем, что выполняется множество стандартных правил (например, формирование новых отношений в соответствии с описанием «inverseOf»). Для нашего примера это необязательно. Поэтому можно на вкладке «OWL 2 RL -> OWL2RL Control» выключить все таблицы стандартных правил. После этого в выведенных аксиомах останутся только аксиомы, полученные с использованием созданного правила.

**Задания**

1 Сформировать минимум 5 правил, выявляющие сомнительные описания различных характеристик.

2 Сформировать правило, которое в зависимости от емкости аккумулятора, тактовой частоты процессора, разрешения экрана и его диагонали будет формировать примерное время автономной работы ноутбука в режимах: игра, работа, ожидание.

## Лабораторные работы 6.

Разработка учебной экспертной системы.

**Вопросы (компетенции, навыки) для освоения:**

7. Изучить примеры построения экспертных систем.
8. Научиться создавать и редактировать базу знаний ЭС.
9. Научиться реализовывать и тестировать базу знаний.

**Теоретическая часть**



Экспертная система (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 имеются связи 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

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.

| 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 |

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.

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,

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

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

## 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 играешь с использованием "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?

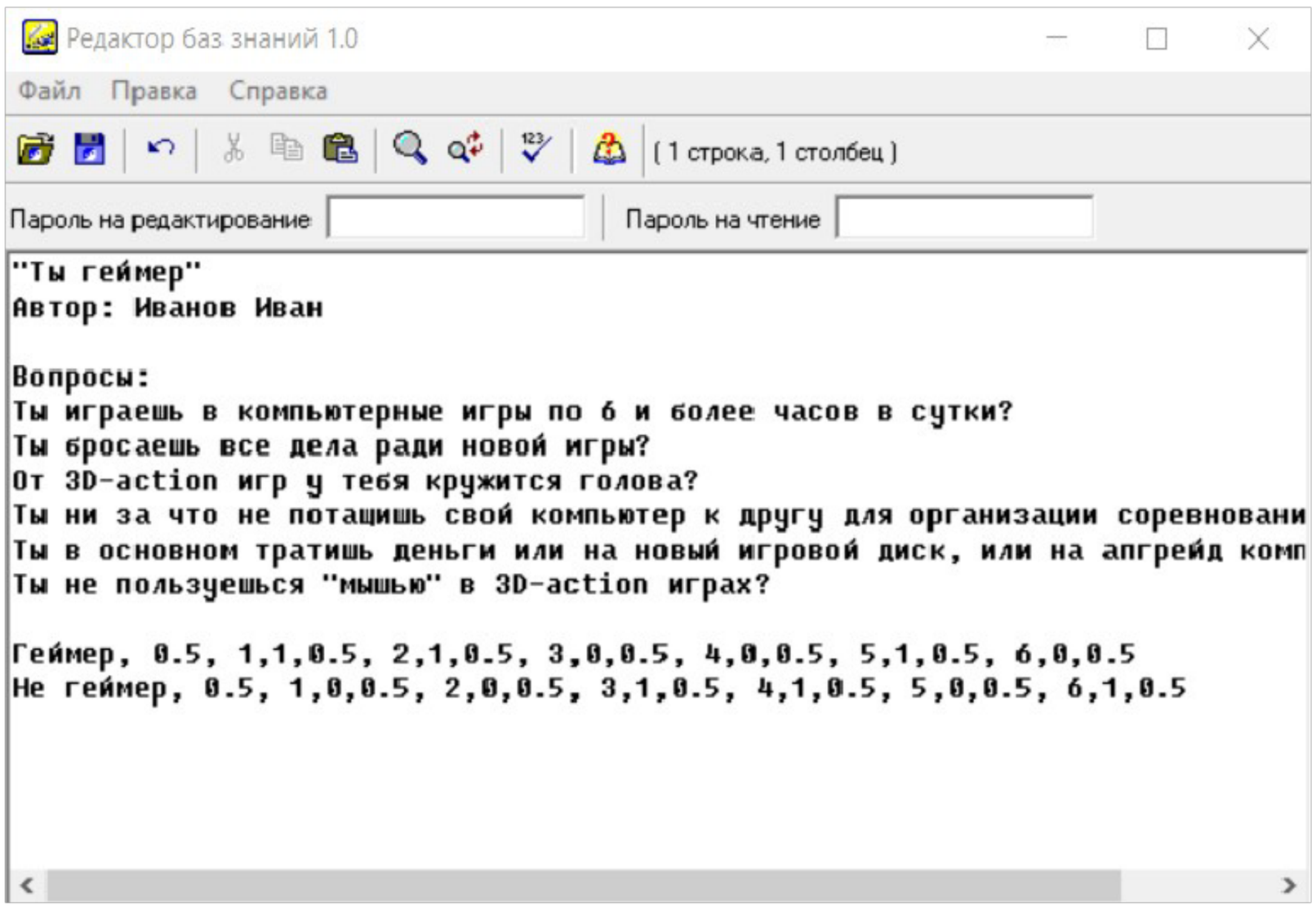
Ты в основном тратишь деньги или на новый игровой диск или на апгрейд компьютера?

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**



На основании рассмотренного примера попробуйте составить собственную БЗ например для диагностики состояния здоровья учеников или анализа обеспеченности литературой учебного процесса

## Лабораторная работа 7.

Анализ существующих образовательных платформенных решений.

**Вопросы (компетенции, навыки) для освоения:**

10. Изучить возможности существующих российских образовательных онлайн-платформ.

11. Научиться анализировать образовательные российские и зарубежные онлайн-платформы.

**Задания для выполнения и методические рекомендации:**

**Задание 1**

Используя приведенный ниже список российских образовательных онлайн-платформ, а также другие интернет-источники составьте сводную таблицу (название платформы, образовательные задачи, предполагаемый возраст учащихся, возможности использования в учебном процессе, преимущества, недостатки).

***Список российских и зарубежных онлайн-платформ***

**Универсариум** (http://universarium.org/) – проект, позиционируемый как российская МООС-платформа. Запущен при поддержке Агентства стратегических инициатив и РИА-Новости. Разрабатывают собственные курсы совместно с российскими вузами и пытаются получить право выдавать сертификаты государственного образца тем, кто успешно проходит курсы. На данный момент три курса уже запущены, готовятся к запуску новые курсы. Есть планы по переводу своих курсов на английский язык и выход на европейскую аудиторию. Курсы позиционируются как отдельные курсы, которые могут быть встроены в бóльшие образовательные программы. Разрабатываются совместно с преподавателями, почти все преподаватели-авторы стартовых курсов – из МГУ им. Ломоносова.



**Eduson** (https://www.eduson.tv/) – платформа для

предоставления образовательных курсов по бизнес-тематикам. Ориентированы на профессионалов. По итогам прохождения курсов можно получить сертификат Eduson. Курсы платные, для корпораций есть вариант подписки на все курсы сразу. Есть бесплатные курсы, но они, как правило, разработаны собственными силами. Платные курсы создаются преподавателями из известных университетов, в основном из США, есть также курсы от российских авторов. Проект нацелен изначально на страны BRIC, стартовал на английском языке, русский язык был добавлен позже.

**Uniweb** (http://uniweb.ru/) – Проект позиционируется как платформа онлайнобучения для распространения как отдельных курсов, так и образовательных программ. Курсы разрабатываются совместно с вузами. Основной партнер – РАНХиГС, также есть курсы с факультетами МГУ, МГИМО, РМОУ. Часть курсов предоставляется бесплатно, большая часть – на платной основе. По итогам прохождения платных курсов выдаются сертификаты от вузов-авторов курсов. Для вузов создание курсов бесплатно, применяется модель "revenue sharing" (часть оплаты за обучение остается у Uniweb)

**Digital October** (coursera.digitaloctober.ru) – Digital October совместно с Coursera пошли по пути перевода курсов Coursera на русский язык. К ним позднее присоединилась Abbyy Software (в части перевода). Один курс запущен (русский перевод "Gamification"), переводятся новые курсы. Также в рамках проекта запущены «открытые лаборатории» - прохождение оригинального курса на Coursera (на английском языке) с помощью российских менторов (профессионалы в изучаемой области).

**Univer.tv** – Медиатека видеозаписей лекций, образовательных фильмов, выступлений на конференциях. Источники видеозаписей – самостоятельно присланное пользователями, лучшее с других ресурсов интернета. В настоящее время ресурс активно не развивается – создатели проекта делают interneturok.ru

**Interneturok.ru** – российский аналог Khan Academy. Самостоятельно записывают видеоуроки для школьников с привлечением как своих, так и сторонних учителей. Пока работают на деньги инвестора, пути монетизации не озвучиваются. Доступ ко всем урокам бесплатный. Основная задача на текущем этапе – повышение качества уроков.

**Intuit.ru** (Национальный открытый университет ИНТУИТ) – Образовательное учреждение, предлагающее две программы бакалавриата и несколько программ второго высшего, большое количество отдельных курсов (повышение квалификации). Все программы проходят целиком онлайн, по итогам программ и курсов выдаются документы установленного образца. Специализируются на IT тематике.

**Хекслет** (hexlet.org) – проект предоставления бесплатных курсов на русском языке. Создан и поддерживается на средства энтузиастов. Основная тематика курсов - IT, есть один курс от преподавателя из РАНХиГС.

**Аргус-М** (www.argusm-edu.ru) – проект, предоставляющий доступ к большому количеству тестов, используемых для проверки знаний. Позиционируется в качестве помощи студентам в подготовке к сдаче экзаменов. Доступ к части тестов платный.

**Университет в кармане** (http://moyuniver.ru/) – мобильное приложение, содержащее базу знаний ответов на вопросы по образовательным программам школы и вуза, тесты для проверки знаний, конструктор эссе.

**Lektorium.tv** – медиатека видеозаписей лекций. Лекции записываются российскими вузами и известными лекториями. Проект стартовал в 2009 году при поддержке Росмолодёжи. За доступ к лекциям плата не берется. Для вузов есть возможность платной записи лекций силами Lektorium.tv. Свои лекции опубликовали 15 вузов (большая часть из Санкт-Петербурга) и 22 лектория. По сути это именно библиотека – каталогизатор видео, выложенных на YouTube

**Businesslearning.ru** – проект позиционируется как “система дистанционного бизнес-образования”. Проект создан совместно «Альянс-Медиа» и «Международный институт менеджмента ЛИНК». В проекте представлены 104 модуля на тематику предпринимательской деятельности. По каждому модулю можно пройти дистанционную аттестацию. Также есть возможность пройти очную аттестацию (в Москве) и получить сертификат государственного образца. Осуществляется при

поддержке программы поддержки предпринимательства Правительства Москвы. С начала 2013 года регистрация на очные аттестации приостановлена.

**Eclass** (http://www.eclass.cc/) – каталогизатор курсов МООС с разных платформ. Данные по новым курсам и по датам старта курсов автоматически обновляются, есть возможность делать подборки курсов.

**Университет без границ** (http://universitetbezgraniz.ru/) – курсы в онлайн режиме на тематику liberal arts. Делают упор на обучение методам исследования. Преподаватели – исследователи из зарубежных и российских университетов. Семинары и часть заданий проходят в онлайн-режиме. Все курсы платные, по итогам выдается сертификат.

**Нетология** (http://netology.ru/) – платформа онлайн-обучения ITспециалистов. Само обучение платное, бесплатно доступны открытые занятия, проходящие в виде вебинаров.

**Учи новое** (http://uchinovoe.ru/) – платформа для размещения коротких курсов на любую тематику. Предоставляет возможность авторам устанавливать платный доступ к курсам и проверять задания обучающихся. Проект пока что находится официально в статусе «бета», однако уже размещено более 70 курсовуроков.

**Get2Know** (http://get2know.ru/) – сервис вебинаров, онлайн-консультаций, онлайн трансляции лекций. За доступ к большинству вебинаров берется плата, которая делится с автором вебинара. Количество доступных вебинаров пока что невелико.



**Zillion** (http://zillion.net/) – «образовательный медиаресурс». Большое количество вебинаров, обучающих статей на тематику ведения бизнеса, реализации идей и личностного роста.

Доступ к некоторым вебинарам платный.

**МФПУ «Синергия»** (http://www.mfpa.ru/r/distance/). Образовательное учреждение, предлагающее онлайн программы высшего образования. Студентам предоставляется доступ к обучающим материалам через портал вуза, а также возможность участия в семинарах через вебинары и консультаций с преподавателями по Skype. При окончании программы, подразумевающей выдачу диплома государственного образца необходимо сдавать государственные экзамены лично в Москве. Обучение платное.

**МЭСИ Онлайн обучение** (http://www.mesi.ru/education/higher/zaochnoe-on-lin/). Образовательное учреждение, предлагающее онлайн программы высшего образования. До 90% обучения проходит в онлайн режиме через вебинары и консультации с преподавателями по Skype. Итоговые аттестационные мероприятия и государственные экзамены сдаются лично в Москве либо в одном из филиалов МЭСИ. Обучение платное.

**LECTA** (https://lecta.rosuchebnik.ru) универсальная образовательная платформа, сочетающие в себе обеспечение учащихся учебными материалами в электронной форме и инструменты реального контроля эффективности процесса обучения.

**ЯКласс** (https://www.yaklass.ru) платформа электронного образования для школ, а также обучающая онлайн-площадка для школьников и их родителей. Технология сайта позволяет проводить электронные тестирования и генерировать задания, уникальные для каждого ученика.

**Uchi.ru** – образовательный портал интерактивного обучения детей при помощи интересных игровых заданий и задач.

## Список литературы


**Основная литература:**

1. Пальмов, С.В.<BR>    Интеллектуальные системы и технологии Электронный ресурс : учебное пособие / С.В. Пальмов. - Самара : Поволжский государственный университет телекоммуникаций и информатики, 2017. - 195 c. - 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 c. — ISBN 978-5-4497-1092-5. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/108282.html

**Дополнительная литература:**

1. Аверченков, В. И. Система формирования знаний в среде Интернет : Монография / Аверченков В. И. - Брянск : Брянский государственный технический университет, 2012. - 181 c. - 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, 2011. – Часть 1. – 175 с. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=208933
3. Павлов, С. И. Системы искусственного интеллекта : учебное пособие / С. И. Павлов. – Томск : Томский государственный университет систем управления и радиоэлектроники, 2011. – Часть 2. – 194 с. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=208939
4. Сергеев, Н. Е. Системы искусственного интеллекта : учебное пособие : [16+] / Н. Е. Сергеев. – Таганрог : Южный федеральный университет, 2016. – Часть 1. – 123 с. : схем., ил., табл. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=493307
5. 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 c. — ISBN 978-5-4497-0868-7. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/102054.html

**Методическая литература**:

1. Методические указания к самостоятельной работе

**Интернет-ресурсы:**

1. Воройский Ф. С. Информатика. Энциклопедический словарь-справочник: введение в современные информационные и телекоммуникационные технологии в терминах и фактах. - М.: ФИЗМАТЛИТ, 2006. - 768 с. – Доступно: http://physics-for-students.ru/bookpc/informatika/slovar.zip
2. Иванов В. Основы искусственного интеллекта – https://libtime.ru/expertsystems/osnovy-iskusstvennogo-intellekta.html
3. Романов П.С. Основы искусственного интеллекта; Учебно-метод. пособие. – http://www.studfiles.ru/preview/2264160/
4. Сайт Основы ИИ – https://sites.google.com/site/osnovyiskusstvennogointellekta/
5. Соболь Б.В. Информатика: учебник/ Б.В. Соболь [и др.] – Изд. 3-е, дополн. – Ростов н/Д : Феникс, 2007. – 446 с. – Доступно: http://physics-for-students.ru/bookpc/informatika/Sobol.rar

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Пятигорский институт (филиал) СКФУ

# Методические указания

по выполнению лабораторных работ
по дисциплине
**«Искусственный интеллект в профессиональной сфере »**
для направления подготовки **10.03.01 Информационная безопасность**
направленность (профиль) **Безопасность компьютерных систем**

**Пятигорск**
**2022**

## СОДЕРЖАНИЕ

## 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. Самостоятельная работа студентов играет значительную роль в рейтинговой технологии обучения.

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).

Самостоятельная работа по дисциплине «Искусственный интеллект в профессиональной сфере » направлена на формирование следующих **компетенций**:

| Индекс | Формулировка: |
|---|---|
| **ПК-3** | Способен адаптировать и модифицировать специализированное программное обеспечение, методы и алгоритмы систем искусственного интеллекта и машинного обучения в профессиональной деятельности |



## 2. Цели и задачи самостоятельной работы

Ведущая цель организации и осуществления СРС совпадает с целью обучения студента – формирование набора общенаучных, профессиональных и специальных компетенций будущего бакалавра по направлению подготовки «Педагогическое образование».

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. Технологическая карта самостоятельной работы студента

| Коды реализуемых компетенций, индикатора(ов) | Вид деятельности студентов | Средства и технологии оценки | Объем часов, в том числе | | |
|---|---|---|---|---|---|
| | | | СРС | Контактная работа с преподавателем | Всего |
| 5 семестр | | | | | |
| ИД-1$_{ПК-3}$ ИД-2$_{ПК-3}$ | Подготовка доклада | Доклад | 9 | 1 | 10 |
| ИД-1$_{ПК-3}$ ИД-2$_{ПК-3}$ | Подготовка к лабораторным занятиям | Собеседование | 7,29 | 0,81 | 8,1 |
| ИД-1$_{ПК-3}$ ИД-2$_{ПК-3}$ | Разработка исследовательского проекта | Защита проекта | 20,16 | 2,24 | 22,4 |
| | | Итого за 5 семестр | 36,45 | 4,05 | 40,5 |

| | | | |
|---|---|---|---|
| Итого | 36,45 | 4,05 | 40,5 |

## 4. Порядок выполнения самостоятельной работы студентом

*4.1. 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, проверить с 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.

*4.2. 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.

## Вопросы для собеседования

Базовый уровень

Тема 1. Применение ИИ в современной жизни.

1. Общее представление об искусственном интеллекте.
2. Использование искусственного интеллекта в различных сферах.
3. Перспективы использования искусственного интеллекта.

Тема 2. Программное обеспечение и онлайн сервисы, применяемые в ИИ.

4. Экспериментальный эволюционный характер разработок систем ИИ.
5. Возможности программных средств с применением ИИ.
6. Требования к программному обеспечению ИИ.
7. Языки программирования для задач ИИ.
8. Понятие об онтологиях. Возможности применения онтологий в профессиональной деятельности.

Тема 3. Методы искусственного интеллекта в современном образовании.

1. Искусственный интеллект в образовании.
2. Искусственный интеллект: конкурент или помощник учителя?
3. «Беспилотное» образование.

Тема 4. Применение искусственного интеллекта при внедрении цифровых образовательных платформенных решений.

4. Интеллектуальные тьюторские системы и адаптивное обучение.
5. Примеры и особенности систем адаптивного обучения.
6. Стандартные задачи адаптивного обучения.
7. Интеллектуальный анализ данных и образовательная аналитика.
8. Характеристики примеры российских образовательных онлайн-платформ.

Повышенный уровень

Тема 1. Применение ИИ в современной жизни.

9. Сферы применения систем искусственного интеллекта.
10. Проблемы использования искусственного интеллекта.
11. Общее представление о применении искусственного интеллекта в профессиональной деятельности.

Тема 2. Программное обеспечение и онлайн сервисы, применяемые в ИИ.

1. Классификация интеллектуальных систем.
2. Языки ЛИСП и ПЛЭНЕР в ИИ.
3. Применение языка программирования PYTHON в разработке систем ИИ.
4. Описание языков R и PROLOG для решения задач ИИ.
5. Возможности системы Protégé для создания онтологий.

Тема 3. Методы искусственного интеллекта в современном образовании.

1. Проблемы современного образования, которые можно решить с помощью ИИ.
2. Проблемы использования искусственного интеллекта в образовании.
3. Методы искусственного интеллекта как способ преодоления фундаментальных проблем современного образования.
4. Характеристика и примеры зарубежных образовательных онлайн-платформ.

Тема 4. Применение искусственного интеллекта при внедрении цифровых образовательных платформенных решений.

1. Применение искусственного интеллекта для оценки качества образовательного процесса.
2. Автоматическая оценка качества письменных работ обучающихся.
3. Анализ обратной связи от обучающихся и контроль процесса обучения на основе обработки текстовой информации из социальных сетей и образовательных форумов.
4. Технологии искусственного интеллекта: возможности и применение в образовательной среде.

*4.4. 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 – решающее условие настоящего творчества, для него наконец-то появляется время. Изначально 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 дисциплины, либо определена самостоятельно, исходя из интересов студента (в рамках изучаемой дисциплины). Выбранную тему необходимо согласовать 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 Times New Roman;
- 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 допускает 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.

*Описание шкалы оценивания*

Максимально возможный балл за весь текущий контроль устанавливается равным 55. Текущее контрольное мероприятие считается сданным, если студент получил за него не менее 60% от установленного для этого контроля максимального балла. Рейтинговый балл, выставляемый студенту за текущее контрольное мероприятие, сданное студентом в установленные графиком контрольных мероприятий сроки, определяется следующим образом:

| Уровень выполнения контрольного задания | Рейтинговый балл (в % от максимального балла за контрольное задание) |
|---|---|
| Отличный | 100 |
| Хороший | 80 |
| Удовлетворительный | 60 |
| Неудовлетворительный | 0 |

**Темы рефератов**

Базовый уровень

1. Возможности и применение ИИ в образовательной среде.
2. Современные программные продукты, основанные на ИИ.
3. Общая характеристика языка программирования PROLOG.
4. Экспертные и интеллектуальные обучающие системы.
5. Модели и алгоритмы распознавания в экспертных системах.
6. Модели представления знаний.
7. Особенности использования семантических сетей для представления знаний.
8. Продукционные базы данных.
9. Фреймовые модели в системах искусственного интеллекта.
10. Онтологии и их применение в профессиональной деятельности.
11. Возможности системы Protégé для создания онтологий.
12. Прогнозирующие интеллектуальные системы.
13. Планирующие интеллектуальные системы.
14. Общий анализ существующих образовательных платформенных решений.

Повышенный уровень

1. Виды экспертных систем.
2. Особенности разработки учебной экспертной системы.
3. Характеристика оболочки экспертных систем CLIPS.
4. Интеллектуальные системы контроля и принятия решений.
5. Понятие и общие характеристики нейронных сетей.
6. Нейронные сети в системах искусственного интеллекта.
7. Области применения нейроинформатики.
8. Редакторы онтологий.
9. Основы RDF и OWL.

10. Байесовская стратегия оценки выводов.
11. Российские образовательные онлайн-платформы.
12. Зарубежные образовательные онлайн-платформы.
13. Специализированные программы для нечетких систем.
14. Возможности программы SciLab для работы с нечеткими базами данных.

*4.5. 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 | 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. |
| «Удовлетворите | Уровень | Обнаруживает | Студент | Проект |

| 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 |
|---|---|---|---|---|
| льно» | 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 соответствующими информационными технологиями | выполнил большую часть возложенной на него работы. | сдан 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 | 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 для выполнения проектных заданий**

1. История развития искусственного интеллекта в России.
2. История развития искусственного интеллекта за рубежом.
3. Искусственный интеллект в профессиональной сфере (в зависимости от направления подготовки).
4. Анализ данных учебной активности обучающихся с применением ИИ.
5. Анализ образовательных данных проектной деятельности.
6. Google Data Studio как средство анализа и визуализации образовательных данных.
7. Разработка онтологии профессиональной предметной области.
8. Принципы разработки систем адаптивного обучения.
9. Зарубежные системы адаптивного обучения.
10. Российские системы адаптивного обучения.
11. Классификация искусственных нейронных сетей.
12. Глубокие искусственные нейронные сети для решения задач планирования.
13. Глубокие искусственные нейронные сети для решения задач прогнозирования.
14. Глубокие искусственные сверточные нейронные сети для решения задач распознавания изображений.
15. Сравнительная характеристика полносвязных и сверточных искусственных нейронных сетей для решения задач классификации и кластеризации.

*4.5. 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 - 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, прочитанный 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.

## 1. Контроль самостоятельной работы студентов

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.

## 2. Список литературы для выполнения СРС

**Основная литература:**

1. Пальмов, С.В.<BR>    Интеллектуальные системы и технологии Электронный ресурс : учебное пособие / С.В. Пальмов. - Самара : Поволжский государственный университет телекоммуникаций и информатики, 2017. - 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 c. — ISBN 978-5-4497-1092-5. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/108282.html

**Дополнительная литература:**

1. Аверченков, В. И. Система формирования знаний в среде Интернет : Монография / Аверченков В. И. - Брянск : Брянский государственный технический университет, 2012. - 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, 2011. – Часть 1. – 175 с. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=208933

3. Павлов, С. И. Системы искусственного интеллекта : учебное пособие / С. И. Павлов. – Томск : Томский государственный университет систем управления и радиоэлектроники, 2011. – Часть 2. – 194 с. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=208939

4. Сергеев, Н. Е. Системы искусственного интеллекта : учебное пособие : [16+] / Н. Е. Сергеев. – Таганрог : Южный федеральный университет, 2016. – Часть 1. – 123 с. : схем., ил., табл. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=493307

5. 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 c. — ISBN 978-5-4497-0868-7. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/109054.html



**Методическая литература**:

1. Методические указания к самостоятельной работе (электронный вариант)

**Интернет-ресурсы:**

1. Воройский Ф. С. Информатика. Энциклопедический словарь-справочник: введение в современные информационные и телекоммуникационные технологии в терминах и фактах. - М.: ФИЗМАТЛИТ, 2006. - 768 с. – Доступно: http://physics-for-students.ru/bookpc/informatika/slovar.zip
2. Иванов В. Основы искусственного интеллекта – https://libtime.ru/expertsystems/osnovy-iskusstvennogo-intellekta.html
3. Романов П.С. Основы искусственного интеллекта; Учебно-метод. пособие. – http://www.studfiles.ru/preview/2264160/
4. Сайт Основы ИИ – https://sites.google.com/site/osnovyiskusstvennogointellekta/
5. Соболь Б.В. Информатика: учебник/ Б.В. Соболь [и др.] – Изд. 3-е, дополн. и перераб. – Ростов н/Д: Феникс, 2007. – 446 с. – Доступно: http://physics-for-students.ru/bookpc/informatika/Sobol.rar