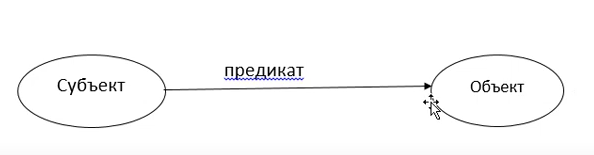
# Что такое онтология

Дескрипционные логики. Все они базируются на одном простом утверждении: любая информация, как бы сложно и запутанно она не звучала естественным языком, может быть сведена без потери смысла к цепочке утверждений, каждое из которых очень простое.



Подлежащее -> сказуемое -> дополнение. Конструкция из трёх элементов называется триплет

**Пример**

**Дано сложно предложение**



**Цепочка простых**

****

<https://www.youtube.com/watch?v=VIpV-hVo3bY&list=PLMDuaURn3ViYuD2HLtf0u4WTTGRFS-dIN>

Для записи онтологии (последовательности триплетов) существует два языка:



RDF – он позволяет просто записывать триплеты и накапливать их в определенном хранилище (имеет практически текстовый формат)

OWL – позволяет кроме фактов выраженных в виде триплетов, задавать определенные логические правила над этим хранилищем. На основе этих логических правил при больших объемах данных можно выявлять некоторые скрытые знания которые невооруженным взглядом не видны, но могут быть вычислены.

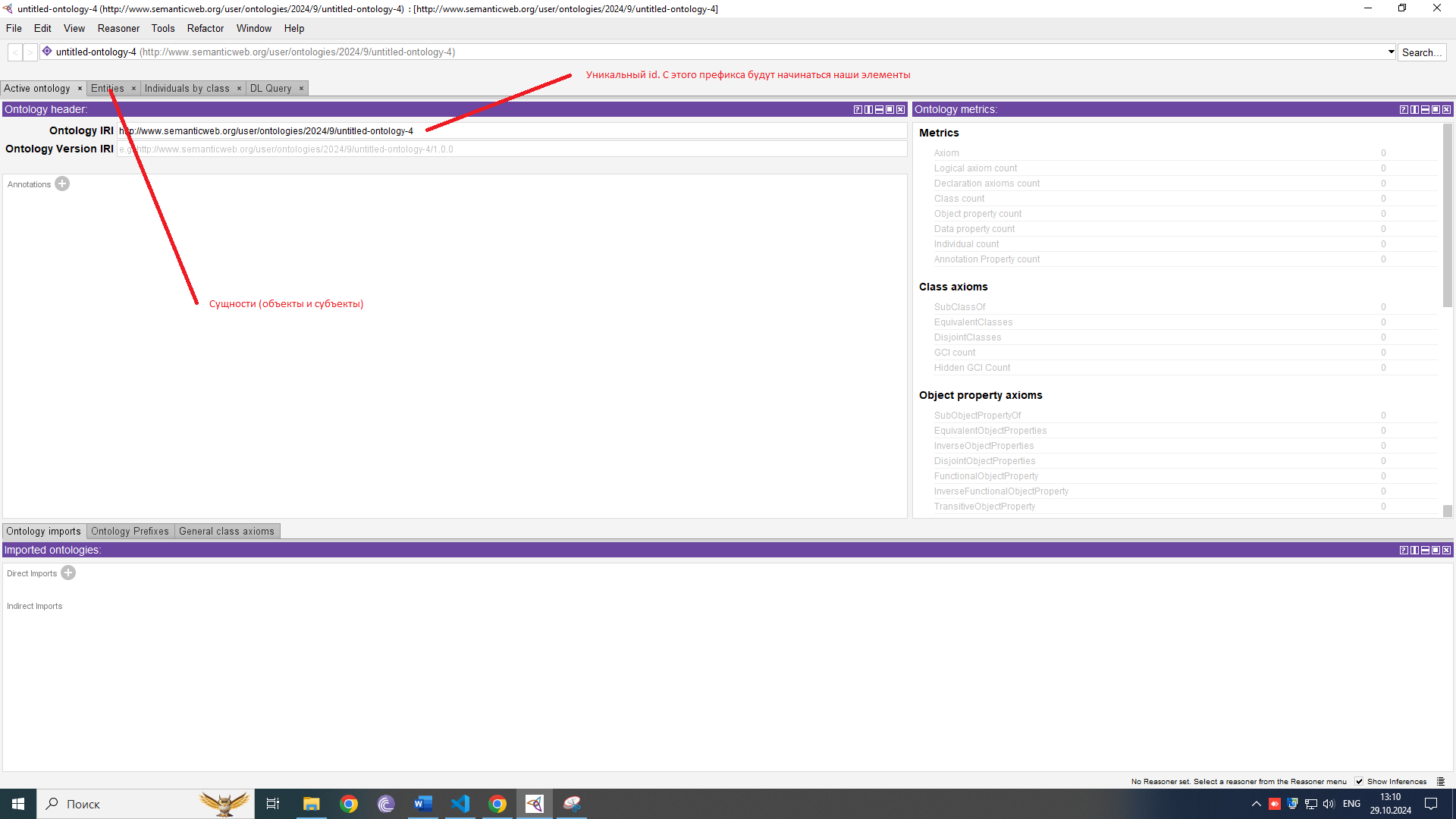
Отличие онтологии от БД: позволяют находить скрытые знания.

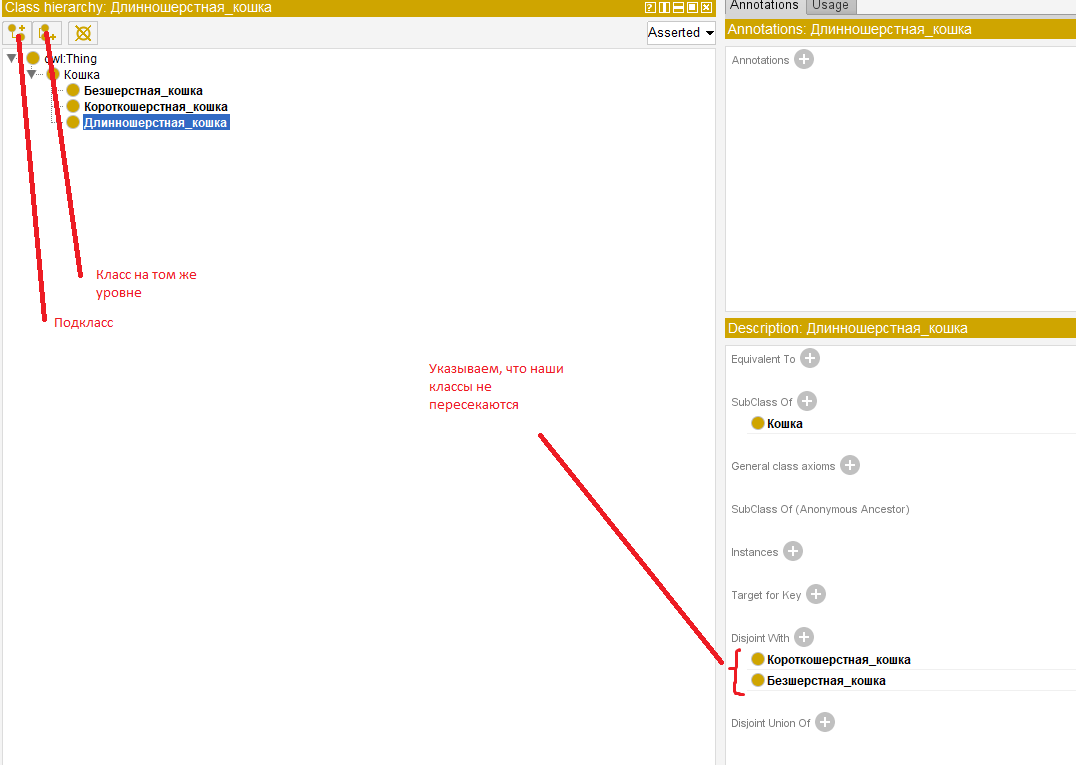
Аксиома об открытости мира

Аксиома о закрытости мира

Обычные БД не основаны на закрытости мира. Это значит, что те данные, которые расположены в реляционных таблицах принимаются как истинные, а любые факты, которые туда не попали считаются ложными!!!

**Reasoner. Как онтология "находит" скрытые знания**

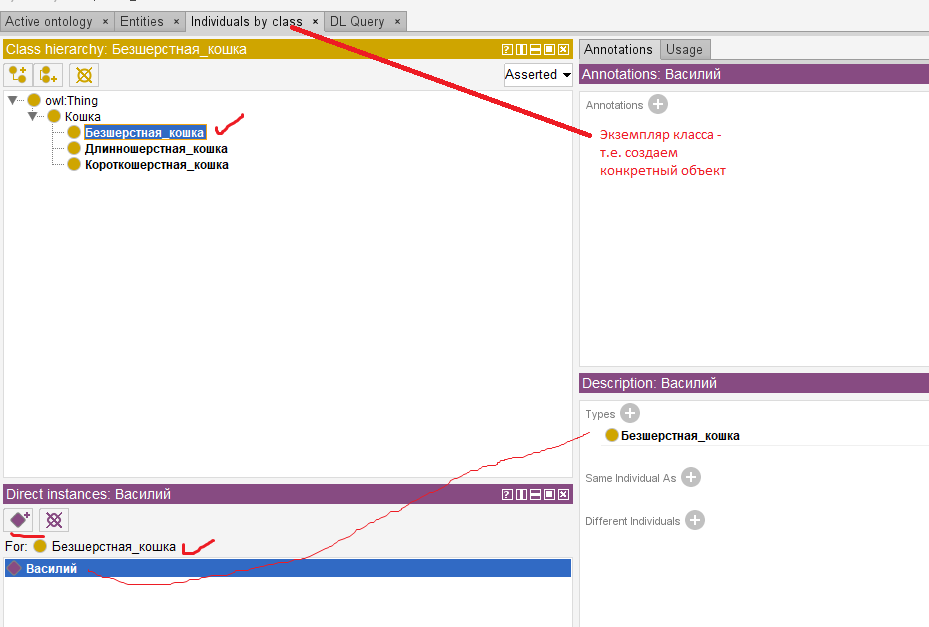




Это триплет!!

Безшерстная\_кошка -> не пересекается -> короткошерстной (субъект ->предикат-> объект)

Безшерстная\_кошка -> не пересекается -> длинношерстной



Создали экземпляр класса. Триплет

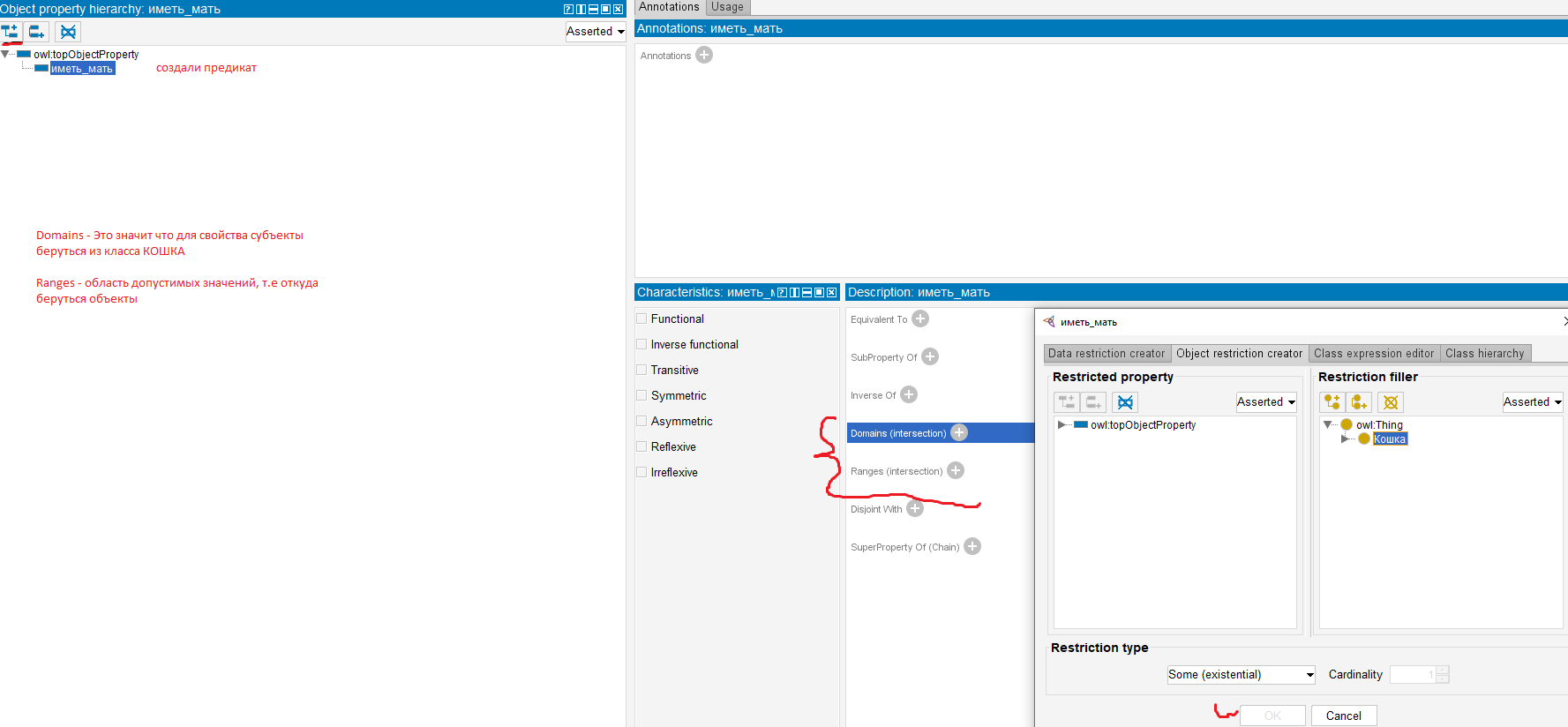
Василий -> имеет тип -> Безшерстная\_Кошка

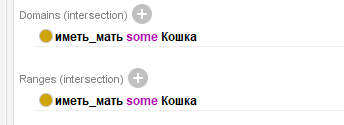
Это предопределенные предикаты “имеет тип”, “не пересекается”

Создание собственного предиката

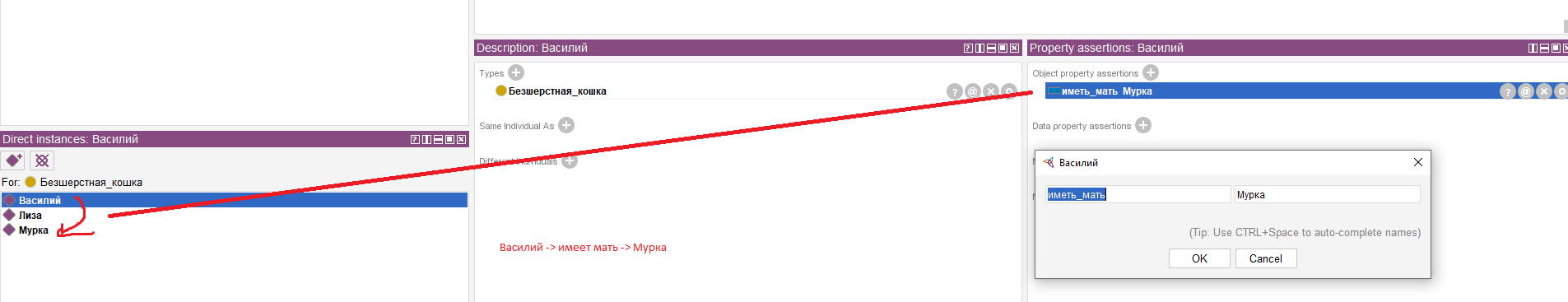
Object properties – это свойства, у которых объектом может выступать уже готовый класс или экземпляр класса

Data properties – это свойство, у которых объектом может выступать простой примитивный тип данных (число, строка)





Теперь можем использовать созданный предикат

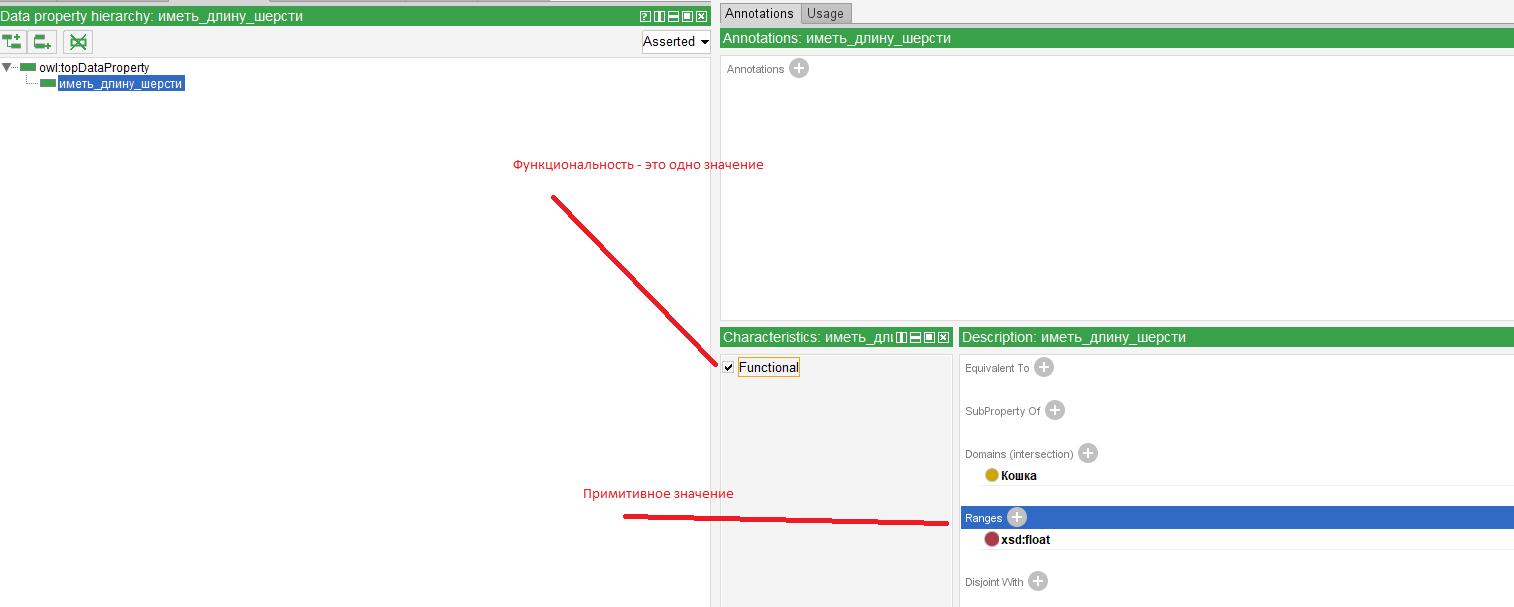


Добавляем еще одну связь иметь\_мать -> Лиза. Запускаем reasoner и видим, что провелся анализ в результате которого появилось предположение что Мурка и Лиза это одна сущность.

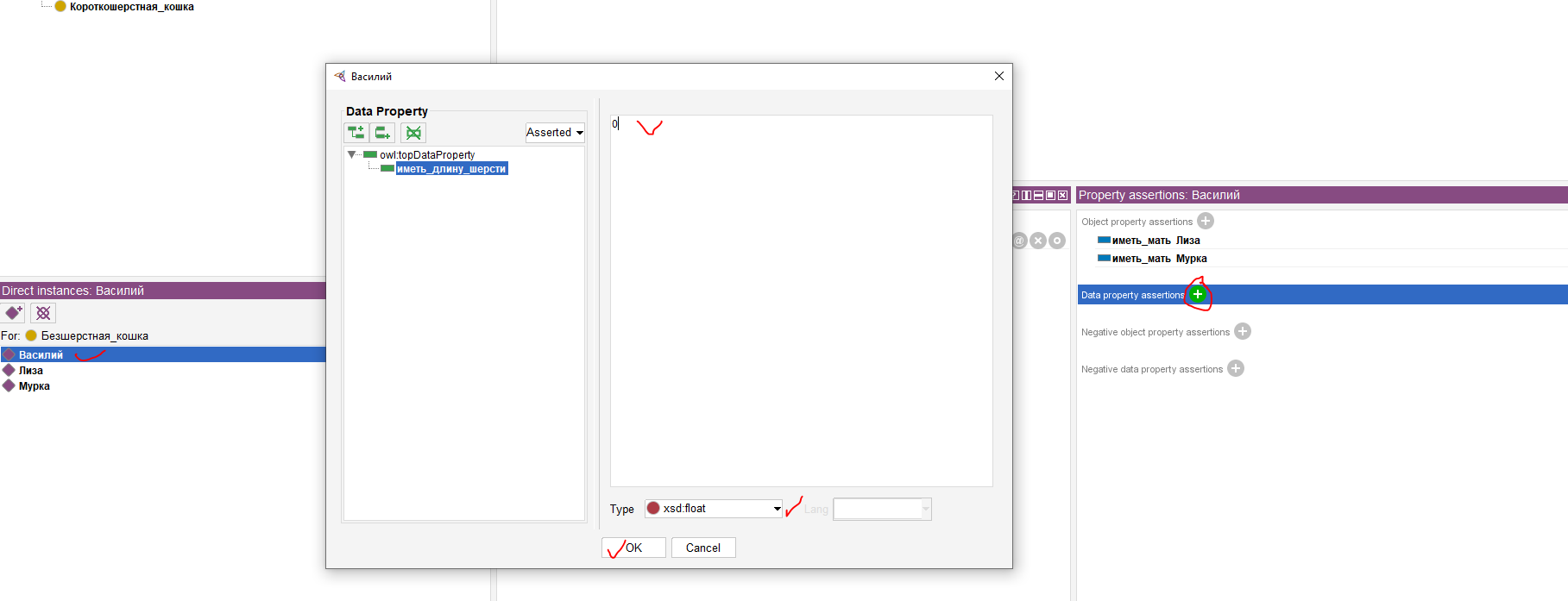
Сохраняем File->export (owl/xml)

# Свойства с примитивными типами (Data Properties)

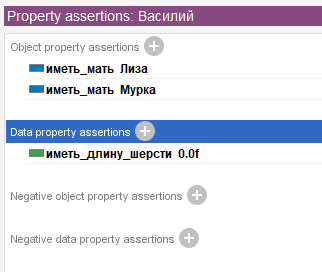
Создаем предикат с учетом того, что объект будет примитивное значение.



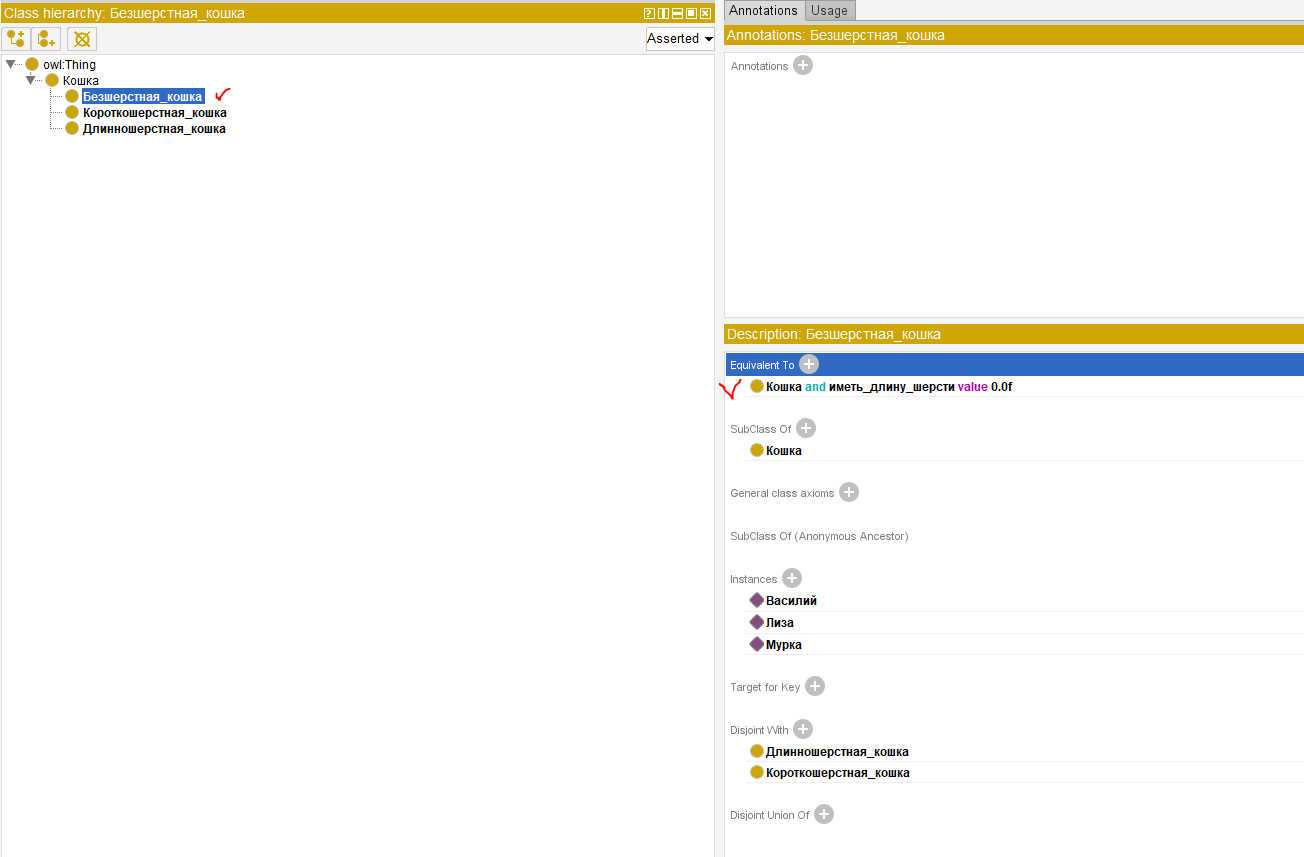
Используем этот предикат



Результат

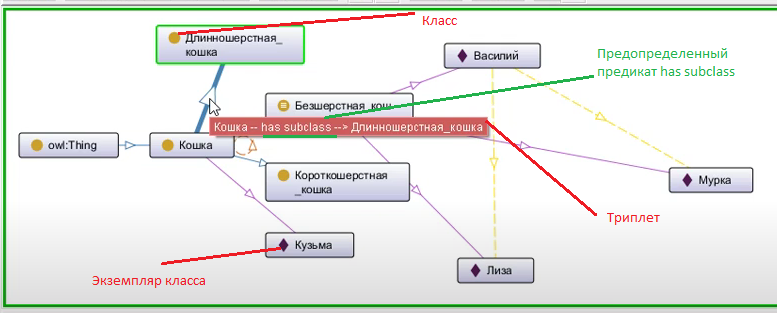


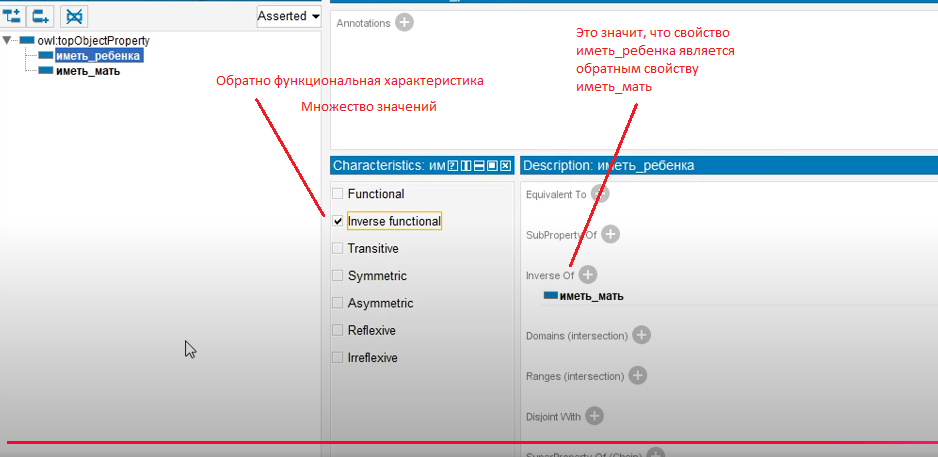
Вкладка Equivalent to – означает что мы можем описать при помощи логических правил какие конкретно экземпляры относятся к тому или иному классу



Описание: должны выполняться два утверждения: экземпляр должен быть КОШКА и это существо должно иметь длину шерсти 0.01f

# Invers Of, Inverse functional, а также OntoGRAF





# Transitive (Транзитивность свойств)

