Lab exercises for Functional Syntax (Week 5)

Task 1

- Open https://www.w3.org/TR/2012/REC-owl2-primer-20121211/
- Go to section 1.2 OWL Syntax
- Ensure that you can see all functional style syntax in this document.

Task 2

Create the following ontology in Protege

- Elephant is a sub-concept of the concept Animal.
- Herbivore is a concept whose members are exactly those animals who eat only plants.
- Adult elephant is a concept whose members are exactly those elephants older than 20.
- Adult elephants might have children who are again elephants.
- Djan is an elephant.
- Djan is 23 years old.
- Djan is the child of Zorba.

Open the Elephant ontology you have created above, save the file using OWL functional syntax (hint: use File Save As), now open the file using Word, and look at the functional syntax.

Task 3: Functional Syntax

Write the following statements using functional-style syntax:
You are given the following declarations (note: not all declarations are given):

Class declaration	Person is a class.	Declaration (Class (:Person))
Class declaration	Student is a class.	Declaration (Class (:Student))
Individual declaration	Martin is an individual.	Declaration (NamedIndividual (:Martin))
Property declaration	hasEmail is a data property.	Declaration (DataProperty (:hasEmail))
Property declaration	isEmployedAt is an object property.	Declaration (ObjectProperty (: isEmployedA))
Property declaration	hasSpouse is an object property.	Declaration (ObjectProperty (:hasSpouse))

Write the following assertions:

1. Student is subclass of Person.
2. Martin is a student.
3. Martin has spouse Lenka.
4. Martin is employed at Supercomputing Center. (note: Supercomputing Center is string type)
5. Martin is employed at Masaryk University.
6. Email of Martin is martin@example.org (Note: martin@example.org is string type)
7. Peter is not spouse of Martin. (hint: use NegativeObjectProperty)
8. Email of Martin is not martin@uow.edu.au

Task 4: Writing class expressions in functional syntax

The following example shows a class Child with four individuals BigBoy, BigGIrl, SmallBoy and SmallGirl.
Question: What does DifferentIndividuals(:BigBoy :BigGirl :SmallBoy :SmallGirl) tell us?

```
Declaration( Class( :Child ) )
  Declaration( NamedIndividual( :BigBoy ) )
  Declaration( NamedIndividual( :BigGirl ) )
  Declaration( NamedIndividual( :SmallBoy ) )
  Declaration( NamedIndividual( :SmallGirl ) )
  ClassAssertion( :Child :BigBoy )
  ClassAssertion( :Child :BigGirl )
  ClassAssertion( :Child :SmallBoy )
  ClassAssertion( :Child :SmallGirl )
  DifferentIndividuals( :BigBoy :BigGirl :SmallBoy :SmallGirl )
```

Your task: Use the above example and materials covered in Week 5 lecture notes, write in functional syntax the following statements:

a). Teenager is equivalent to a person whose ages is between 12 and 20.

b). Ivan, Martin and Lenka are individuals of class Person. Ivan hasParent Martin and Ivan hasParent Lenka.

c). hasID is a unique identifier for Person. Martin hasID “5648” and Lenka hasID “1234”. (note: hasID is a data property and not object property)