

## 2η Σειρά Ασκήσεων

(Παράδοση 25/11/2022)

1. Δίνεται η παρακάτω πρόταση φυσικής γλώσσας:

*Η Ιλιάδα συνθέθηκε απο ποιητή που έζησε τον 8ο αιώνα π.Χ. στην Ιωνία της Μικράς Ασίας.*

- Γράψτε την πρόταση σε RDF τριπλέτες **χωρίς κενούς κόμβους**, χρησιμοποιώντας τουλάχιστον 3 τριπλέτες.
- Γράψτε την πρόταση σε RDF τριπλέτες χρησιμοποιώντας **τουλάχιστον έναν κενό κόμβο**, χρησιμοποιώντας τουλάχιστον 3 τριπλέτες.

Να ελέγξετε το RDF σας με τον validator '<http://www.w3.org/RDF/Validator/>' επιλέγοντας να δείχνει και τα triples και τους γράφους και να τα παραθέσετε στην αναφορά.

2. Γράψτε την ακόλουθη πρόταση σε RDF χρησιμοποιώντας *reification*:

*Η google αναφέρει ότι το Τμήμα Η/Υ & Πληροφορικής βρίσκεται στο Ρίο.*

Και εδώ να ελέγξετε το RDF σας με τον validator και να παραθέσετε στην αναφορά τις τριπλέτες και τον γράφο.

3. Γράψτε μια **RDFS** οντολογία σχετικά με **Persons** και **Departments**. Συγκεκριμένα δημιουργήστε:

- **6 κλάσεις(classes)** που θα αναπαριστούν **Person**, **Student**, **Professor**, **Department**, **Lesson**, **Classroom**
- **2 subClassOf** statements, έτσι ώστε το Professor και το Student να είναι υποκλάσεις (subclasses) του Person
- **9 ιδιότητες(properties).**
  - Μια ιδιότητα **has\_name** με περιορισμό το domain να είναι **Person** και το range **literal**. [Ονοματεπώνυμο]
  - Μια ιδιότητα **has\_phone** με περιορισμό το domain να είναι **Person** και το range **literal**.
  - Μια ιδιότητα **has\_email** με περιορισμό το domain να είναι **Person** και το range **literal**.
  - Μια ιδιότητα **has\_age** με περιορισμό το domain να είναι **Person** και το range **integer**.
  - Μια ιδιότητα **member\_of** με περιορισμό το domain να είναι **Person** και το range **Department**.
  - Μια ιδιότητα **teaches** με περιορισμό το domain να είναι **Professor** και το range **Lesson**
  - Μια ιδιότητα **les\_name** με περιορισμό το domain να είναι **Lesson** και το range **literal** [Όνομα Μαθήματος].
  - Μια ιδιότητα **taught\_by** με περιορισμό το domain να είναι **Lesson** και το range **Professor**
  - Μια ιδιότητα **dep\_name** που αναπαριστά το όνομα ενός τμήματος με περιορισμό το domain να είναι **Department** και το range **literal**.
  - Μια ιδιότητα **dep\_city** που αναπαριστά την πόλη ενός τμήματος με περιορισμό το domain να είναι **Department** και το range **literal**.
  - Μια ιδιότητα **room\_name** με περιορισμό το domain να είναι **Classroom** και το range **literal** [όνομα αίθουσας]
  - Μια ιδιότητα **room\_capacity** με περιορισμό το domain να είναι **Classroom** και το range **integer** [χωρητικότητα αίθουσας]

xiii. Μια ιδιότητα **room\_department** με περιορισμό το domain να είναι **Classroom** και το range **Department** [τμήμα που ανήκει η αίθουσα]

- Δημιουργήστε τουλάχιστον 6 **τμήματα** από τα οποία τα 3 θα πρέπει να βρίσκονται [dep\_city] στην πόλη της Πάτρας και τα υπόλοιπα σε άλλες Πόλεις
- Δημιουργήστε τουλάχιστον 10 **καθηγητές** και 20 **φοιτητές** συνολικά και ορίστε τιμές για τις ιδιότητες τους. Στην δήλωση τους θα αναφέρετε την κλάση που ανήκουν [καθηγητής ή φοιτητής] ωστόσο ΔΕΝ πρέπει να αναφέρετε ότι ανήκουν στην κλάση Person.
- Δημιουργήστε 15 **μαθήματα** [Lesson] συνολικά ορίζοντας τιμή στις ιδιότητες *lec\_name* και *taught\_by*. ΔΕΝ θα πρέπει να δηλώσετε ότι ανήκουν στην κλάση Lesson.
- Δημιουργήστε τουλάχιστον 3 αίθουσες **για κάθε τμήμα**.

4. Σχηματίστε τα παρακάτω **SPARQL** ερωτήματα για την οντολογία που σχηματίσατε:

- i. Το πρώτο επιστρέφει τα τηλέφωνα όλων των καθηγητών
- ii. Το δεύτερο επιστρέφει τα τηλέφωνα όλων των μαθητών που έχουν ηλικία μεγαλύτερη από 23 (με χρήση του FILTER)
- iii. Το τρίτο επιστρέφει τα ονοματεπώνυμα όλων των Ατόμων [πρέπει να ανήκουν στην κλάση Persons] που είναι μέλη σε τμήμα που βρίσκεται στην Πάτρα
- iv. Επιστρέφει τα ονόματα των αιθουσών με χωρητικότητα μεγαλύτερη από 150 που βρίσκονται σε τμήματα στην Πάτρα.

Δοκιμάστε τα ερωτήματα στο RDF αρχείο του ερωτήματος 3, χρησιμοποιώντας μια μηχανή SPARQL ερωτημάτων όπως η ARQ (περιέχεται και στο JenaAPI)

Για κάθε ερώτημα παραθέστε τα αποτελέσματα που επέστρεψε και σχολιάστε αν επέστρεψε τα δεδομένα που περιμένατε. Χρησιμοποιεί η μηχανή SPARQL ερωτημάτων που επιλέξατε κάποιο Μηχανισμό Συμπερασμού [Inference Engine]; Αν ναι αναφέρετε ποιά από τα αποτελέσματα προέκυψαν λόγω εφαρμογής του Μηχανισμού Συμπερασμού.

Οδηγίες για χρήση της ARQ στις διαφάνειες του φροντιστηρίου.

[http://jena.apache.org/tutorials/sparql\\_query1.html](http://jena.apache.org/tutorials/sparql_query1.html)  
<http://www.w3.org/wiki/SparqlImplementations>

5. Αναπτύξετε πρόγραμμα σε γλώσσα Java χρησιμοποιώντας το JenaAPI:

Το πρόγραμμα θα φορτώνει το **rdf** αρχείο που δημιουργήσατε στο ερώτημα 3 και:

**A.** Θα εντοπίζει τα Τμήματα που έχουν οριστεί στο αρχείο RDF και θα δίνει στον χρήστη την δυνατότητα να επιλέξει ένα από αυτά.

Όταν ο χρήστης επιλέξει ένα τμήμα, τότε θα του εμφανίζει πίνακες με το προσωπικό, τους φοιτητές, τις αίθουσες και τα μαθήματα του τμήματος. Για παράδειγμα, για το προσωπικό θα γίνεται παρουσίαση αναλυτικού πίνακα με την παρακάτω μορφή:

*Ονοματεπώνυμο | Ηλικία | Τηλέφωνο*

*Αντίστοιχα για τους φοιτητές, τις αίθουσες και τα μαθήματα του τμήματος.*

**B.** Θα δίνει την δυνατότητα στον χρήστη να προσθέτει νέα δεδομένα (Καθηγητές, Φοιτητές, Τμήματα, Μαθήματα και Αίθουσες) στο RDFαρχείο.

**Γ.** Θα δίνει την δυνατότητα στον χρήστη να δώσει το *URI* ενός *Resource* και θα εμφανίζει όλες τις τριπλέτες [Statements] που υπάρχουν για αυτό στον γράφο. Π.χ: Αν δώσει **URI**: *http://univ/Papadopoulos* θα εμφανίζεται:

<a href="http://univ/Papadopoulos">http://univ/Papadopoulos</a>	<a href="http://univ#has_name">http://univ#has_name</a>	Kostas Papadopoulos
<a href="http://univ/Papadopoulos">http://univ/Papadopoulos</a>	<a href="http://univ#has_age">http://univ#has_age</a>	12
<a href="http://univ/Papadopoulos">http://univ/Papadopoulos</a>	<a href="http://univ#has_phone">http://univ#has_phone</a>	24587652322
<a href="http://univ/Papadopoulos">http://univ/Papadopoulos</a>	<a href="http://univ#member_of">http://univ#member_of</a>	<a href="http://univ/ceid">http://univ/ceid</a>
<a href="http://univ/Papadopoulos">http://univ/Papadopoulos</a>	<a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#type">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</a>	<a href="http://univ/Student">http://univ/Student</a>
<a href="http://univ/Papadopoulos">http://univ/Papadopoulos</a>	<a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#type">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</a>	<a href="http://univ/Person">http://univ/Person</a>

Για κάθε μαθητή ή καθηγητή θα πρέπει να εμφανίζεται και η πληροφορία ότι είναι άτομο [Person] παρόλο που κάτι τέτοιο δεν θα έχει οριστεί άμεσα από εσάς στο RDF αρχείο. Όμοια για κάθε μάθημα θα πρέπει επίσης να εμφανίζεται και η πληροφορία ότι είναι μάθημα παρόλο που δεν θα έχει οριστεί άμεσα από εσάς στο RDF αρχείο.

Για να το πετύχετε αυτό εφαρμόστε ένα μοντέλο Συμπερασμού-Inference [InfModel].

Προτείνεται η χρήση Java με το JenaAPI, αλλά μπορείτε να χρησιμοποιήσετε οποιαδήποτε γλώσσα (C++, C#, ASP, JSP, PHP, Python...) και περιβάλλον προτιμάτε εφόσον διαθέτει δυνατότητα χειρισμού RDF/RDFS μοντέλων και μηχανή για SPARQL ερωτήματα. Καλό είναι να επιλέξετε γλώσσα και αντίστοιχο API που να υποστηρίζει μηχανισμό συμπερασμού(Inference) και διαχείριση οντολογίας OWL ώστε να μπορείτε να το χρησιμοποιήσετε και στο Project του μαθήματος.

Οδηγίες για χρήση του JENA API βρίσκονται στις διαφάνειες του φροντιστηρίου

## **ΣΥΝΔΕΣΜΟΙ**

### **RDF with Jena**

- [http://jena.apache.org/tutorials/rdf\\_api.html](http://jena.apache.org/tutorials/rdf_api.html)
- [http://jena.apache.org/getting\\_started/index.html](http://jena.apache.org/getting_started/index.html)
- <http://krishnakumar15.blogspot.gr/2010/12/integrating-jena-api-with-netbeans-ide.html>
- <http://www.ibm.com/developerworks/xml/library/j-jena/>
- <http://etutorials.org/Misc/Practical+resource+description+framework+rdf/Chapter+8.+Jena+RD+in+Java/>
- [http://rad.ihu.edu.gr/fileadmin/labsfiles/knowledge\\_management/TUTORIALS/Jena.pdf](http://rad.ihu.edu.gr/fileadmin/labsfiles/knowledge_management/TUTORIALS/Jena.pdf)
- [Μελετήστε τα παραδείγματα που περιέχονται στο JENA API στον φάκελο: apache-jena-2.7.4\src-examples\jena\examples\rdf](#)

### **SPARQL with Jena**

- <http://www.ibm.com/developerworks/xml/library/j-sparql/>
- <http://jena.apache.org/tutorials/sparql.html>
- [Μελετήστε τα παραδείγματα που περιέχονται στο JENA API: στον φάκελο: apache-jena-2.7.4\src-examples\larq\examples](#)

### **INFERENCE with Jena**

- <http://jena.apache.org/documentation/inference/index.html>

## **ΠΑΡΑΔΟΤΕΑ:**

Ένα αρχείο **rar** ή **zip** που θα περιέχει:

- Αναφορά σε ηλεκτρονική μορφή(**pdf**) που να εξηγεί αναλυτικά πως αντιμετωπίσατε το κάθε ερώτημα και θα αναλύει -περιέχει τα εξής:
  - Απαντήσεις στα ερωτήματα 1-2.
  - Αναλυτική περιγραφή του RDFαρχείου που δημιουργήσατε για το ερώτημα 3
  - Παράθεση και επεξήγηση των ερωτημάτων SPARQL του ερωτήματος 4 και των αποτελεσμάτων που επέστρεψαν.
  - Περιγραφή της υλοποίησης σας στο ερώτημα 5, σχολιασμό του κώδικα που υλοποιεί τα ζητούμενα και παράθεση εικόνων από την εκτέλεση του για κάθε λειτουργία του.
- Αρχείο **RDF** του ερωτήματος 3
- **Πηγαίο** (κώδικας και project αρχεία αν υπάρχουν) και **εκτελέσιμο** κώδικα για το ερώτημα 5

## **ΤΡΟΠΟΣ ΠΑΡΑΔΟΣΗΣ:**

Η παράδοση του αρχείου θα γίνει ηλεκτρονικά στο site του μαθήματος στο eclass. Η άσκηση είναι ατομική και το αρχείο που παραδίδεται πρέπει να έχει συγκεκριμένη δομή ονόματος: KRWEB\_AM\_ASK\_2.rar

πχ: **KRWEB\_9854\_ASK\_2.rar** (Με λατινικούς χαρακτήρες)

### **ΒΑΘΜΟΛΟΓΗΣΗ:**

Η άσκηση συμμετέχει σε ποσοστό 10% στον τελικό βαθμό του μαθήματος.

Ερωτήματα 1-2: 10%

Ερωτήματα 3-4 : 60%

Ερώτημα 5: 30%