

# **ΑΝΑΠΑΡΑΣΤΑΣΗ ΓΝΩΣΗΣ ΣΤΟΝ ΠΑΓΚΟΣΜΙΟ ΙΣΤΟ**

**2<sup>η</sup> Σειρά Ασκήσεων**

**Ευάγγελος Καψουλάκης**

**1047062**

# Ερώτημα 1

α)

Εικόνες από το <https://www.w3.org/RDF/Validator/>

## The original RDF/XML document

```
1: <rdf:RDF
2:   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3:   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
4:   xmlns:il="http://www.mydomain.org/">
5:
6:   <rdf:Description rdf:about="http://www.mydomain.org/Iliada">
7:     <il:composed_from rdf:resource="http://www.mydomain.org/Poet"/>
8:   </rdf:Description>
9:
10:  <rdf:Description rdf:about="http://www.mydomain.org/Poet">
11:    <il:lived_in rdf:resource="http://www.mydomain.org/Ionia"/>
12:    <il:lived_at>8th century BC</il:lived_at>
13:  </rdf:Description>
14:
15:  <rdf:Description rdf:about="http://www.mydomain.org/Ionia">
16:    <il:located_at>Asia Minor</il:located_at>
17:  </rdf:Description>
18:
19:
20: </rdf:RDF>
```

RDF τριπλέτες:

### Triples of the Data Model

Number	Subject	Predicate	Object
1	<a href="http://www.mydomain.org/Iliada">http://www.mydomain.org/Iliada</a>	<a href="http://www.mydomain.org/composed_from">http://www.mydomain.org/composed_from</a>	<a href="http://www.mydomain.org/Poet">http://www.mydomain.org/Poet</a>
2	<a href="http://www.mydomain.org/Poet">http://www.mydomain.org/Poet</a>	<a href="http://www.mydomain.org/lived_in">http://www.mydomain.org/lived_in</a>	<a href="http://www.mydomain.org/Ionia">http://www.mydomain.org/Ionia</a>
3	<a href="http://www.mydomain.org/Poet">http://www.mydomain.org/Poet</a>	<a href="http://www.mydomain.org/lived_at">http://www.mydomain.org/lived_at</a>	"8th century BC"
4	<a href="http://www.mydomain.org/Ionia">http://www.mydomain.org/Ionia</a>	<a href="http://www.mydomain.org/located_at">http://www.mydomain.org/located_at</a>	"Asia Minor"

Γράφος:

Graph of the data model



β) Με τουλάχιστον έναν κενό κόμβο το RDF :

### The original RDF/XML document

```
1: <rdf:RDF
2:   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3:   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
4:   xmlns:il="http://www.mydomain.org/">
5:
6:   <rdf:Description rdf:about="http://www.mydomain.org/Iliada">
7:     <il:composed_by>
8:       <rdf:Description rdf:nodeID="poet">
9:         <il:lived_in rdf:resource="http://www.mydomain.org/Ionia"/>
10:        <il:lived_at>8th century BC</il:lived_at>
11:      </rdf:Description>
12:    </il:composed_by>
13:  </rdf:Description>
14:
15:  <rdf:Description rdf:about="http://www.mydomain.org/Ionia">
16:    <il:located_at>Asia Minor</il:located_at>
17:  </rdf:Description>
18:
19:
20: </rdf:RDF>
```

Τριπλέτες:

### Triples of the Data Model

Number	Subject	Predicate	Object
1	<a href="http://www.mydomain.org/Iliada">http://www.mydomain.org/Iliada</a>	<a href="http://www.mydomain.org/composed_by">http://www.mydomain.org/composed_by</a>	genid:Upoet
2	genid:Upoet	<a href="http://www.mydomain.org/lived_in">http://www.mydomain.org/lived_in</a>	<a href="http://www.mydomain.org/Ionia">http://www.mydomain.org/Ionia</a>
3	genid:Upoet	<a href="http://www.mydomain.org/lived_at">http://www.mydomain.org/lived_at</a>	"8th century BC"
4	<a href="http://www.mydomain.org/Ionia">http://www.mydomain.org/Ionia</a>	<a href="http://www.mydomain.org/located_at">http://www.mydomain.org/located_at</a>	"Asia Minor"

Γράφος:

Graph of the data model



## Ερώτημα 2

Με χρήση reification:

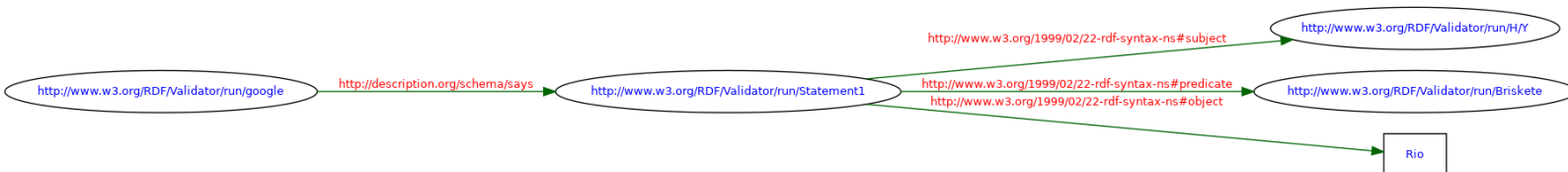
### The original RDF/XML document

```
1: <?xml version="1.0"?>
2: <rdf:RDF
3:   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4:   xmlns:a="http://description.org/schema/">
5:
6:   <rdf:Description rdf:about="Statement1">
7:     <rdf:subject rdf:resource="H/Y" />
8:     <rdf:predicate rdf:resource="Briskete" />
9:     <rdf:object>Rio</rdf:object>
10:  </rdf:Description>
11:
12:   <rdf:Description rdf:about="google">
13:     <a:says rdf:resource="Statement1"/>
14:   </rdf:Description>
15:
16: </rdf:RDF>
```

### Triples of the Data Model

Number	Subject	Predicate	Object
1	<a href="http://www.w3.org/RDF/Validator/run/Statement1">http://www.w3.org/RDF/Validator/run/Statement1</a>	<a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#subject">http://www.w3.org/1999/02/22-rdf-syntax-ns#subject</a>	<a href="http://www.w3.org/RDF/Validator/run/H/Y">http://www.w3.org/RDF/Validator/run/H/Y</a>
2	<a href="http://www.w3.org/RDF/Validator/run/Statement1">http://www.w3.org/RDF/Validator/run/Statement1</a>	<a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#predicate">http://www.w3.org/1999/02/22-rdf-syntax-ns#predicate</a>	<a href="http://www.w3.org/RDF/Validator/run/Briskete">http://www.w3.org/RDF/Validator/run/Briskete</a>
3	<a href="http://www.w3.org/RDF/Validator/run/Statement1">http://www.w3.org/RDF/Validator/run/Statement1</a>	<a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#object">http://www.w3.org/1999/02/22-rdf-syntax-ns#object</a>	"Rio"
4	<a href="http://www.w3.org/RDF/Validator/run/google">http://www.w3.org/RDF/Validator/run/google</a>	<a href="http://description.org/schema/says">http://description.org/schema/says</a>	<a href="http://www.w3.org/RDF/Validator/run/Statement1">http://www.w3.org/RDF/Validator/run/Statement1</a>

### Graph of the data model



## Ερώτημα 3

Δημιουργία των 6 κλάσεων που η κλάση Professor και Student είναι υποκλάσεις της κλάσης Person. Παράθεση εικόνας από το RDF αρχείο.

```
1 <rdf:RDF
2   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
4   xmlns:uni="http://www.mydomain.org/">
5   <rdfs:Class rdf:about="http://www.mydomain.org/Person"/>
6   <rdfs:Class rdf:about="http://www.mydomain.org/Student">
7     <rdfs:subClassOf rdf:resource="http://www.mydomain.org/Person"/>
8   </rdfs:Class>
9   <rdfs:Class rdf:about="http://www.mydomain.org/Professor">
10    <rdfs:subClassOf rdf:resource="http://www.mydomain.org/Person"/>
11  </rdfs:Class>
12  <rdfs:Class rdf:about="http://www.mydomain.org/Department"/>
13  <rdfs:Class rdf:about="http://www.mydomain.org/Lesson"/>
14  <rdfs:Class rdf:about="http://www.mydomain.org/Classroom"/>
```

Δημιουργία των ιδιοτήτων. (Ενδεικτικό screenshot)

```
<rdf:Property rdf:about="http://www.mydomain.org/has_name">
  <rdfs:domain rdf:resource="http://www.mydomain.org/Person"/>
  <rdfs:range rdf:resource="https://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:about="http://www.mydomain.org/has_phone">
  <rdfs:domain rdf:resource="http://www.mydomain.org/Person"/>
  <rdfs:range rdf:resource="https://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:about="http://www.mydomain.org/has_email">
  <rdfs:domain rdf:resource="http://www.mydomain.org/Person"/>
  <rdfs:range rdf:resource="https://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:about="http://www.mydomain.org/has_age">
  <rdfs:domain rdf:resource="http://www.mydomain.org/Person"/>
  <rdfs:range rdf:resource="https://www.w3.org/2000/01/rdf-schema#Integer"/>
</rdf:Property>
<rdf:Property rdf:about="http://www.mydomain.org/member_of">
  <rdfs:domain rdf:resource="http://www.mydomain.org/Person"/>
  <rdfs:range rdf:resource="http://www.mydomain.org/Department"/>
</rdf:Property>
<rdf:Property rdf:about="http://www.mydomain.org/teaches">
  <rdfs:domain rdf:resource="http://www.mydomain.org/Professor"/>
  <rdfs:range rdf:resource="http://www.mydomain.org/Lesson"/>
</rdf:Property>
<rdf:Property rdf:about="http://www.mydomain.org/les_name">
  <rdfs:domain rdf:resource="http://www.mydomain.org/Lesson"/>
  <rdfs:range rdf:resource="https://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
```

Στην αρχή έγινε χρήση του **rdf:id** για την δήλωση των κλάσεων και των ιδιοτήτων αλλά παρατηρήθηκε ότι έβγαζε warnings στο ερώτημα 5 και έγινε αλλαγή σε **rdf:about**.

Στις ιδιότητες όταν το resource είναι μια κλάση χρησιμοποιώ το uni uri ενώ όταν είναι literal ή integer γίνετε χρήση του rdfs uri

Για την δημιουργία των 6 resources για τα departments.

Στο **rdf:type** γίνετε δήλωση ότι ανήκουν στη κλάση Department

Στο **<uni:dep\_city>** η ιδιότητα **dep\_city**. 3 στη Πάτρα 3 αλλού.

Στο **<uni:dep\_name>** η ιδιότητα **dep\_name** δηλώνετε το όνομα του Department.

```
<rdf:Description rdf:about="http://www.mydomain.org/CEID">
  <rdf:type rdf:resource="uni:Department"/>
  <uni:dep_city>Patra</uni:dep_city>
  <uni:dep_name>CEID</uni:dep_name>
</rdf:Description>

<rdf:Description rdf:about="http://www.mydomain.org/chemeng">
  <rdf:type rdf:resource="uni:Department"/>
  <uni:dep_city>Patra</uni:dep_city>
  <uni:dep_name>Chemical Engineering</uni:dep_name>
</rdf:Description>

<rdf:Description rdf:about="http://www.mydomain.org/philosophy">
  <rdf:type rdf:resource="uni:Department"/>
  <uni:dep_city>Patra</uni:dep_city>
  <uni:dep_name>Philosophy</uni:dep_name>
</rdf:Description>

<rdf:Description rdf:about="http://www.mydomain.org/ASFA">
  <rdf:type rdf:resource="uni:Department"/>
  <uni:dep_city>Athens</uni:dep_city>
  <uni:dep_name>Fine Arts</uni:dep_name>
</rdf:Description>

<rdf:Description rdf:about="http://www.mydomain.org/MATH">
  <rdf:type rdf:resource="uni:Department"/>
  <uni:dep_city>Thessaloniki</uni:dep_city>
  <uni:dep_name>Math</uni:dep_name>
</rdf:Description>

<rdf:Description rdf:about="http://www.mydomain.org/UTH">
  <rdf:type rdf:resource="uni:Department"/>
  <uni:dep_city>Volos</uni:dep_city>
  <uni:dep_name>Paedagogical</uni:dep_name>
</rdf:Description>
```

Για τη δημιουργία των καθηγητών και των φοιτητών (Ενδεικτικό screenshot)

Στο **rdf:type** φαίνεται μόνο η κλάση Professor και Student ανάλογα χωρίς να δείχνει ότι είναι στη κλάση Person

```
<rdf:Description rdf:about="http://www.mydomain.org/Professor10">
  <rdf:type rdf:resource="uni:Professor"/>
  <uni:has_name>Professor10</uni:has_name>
  <uni:has_email>email10@uth.uvolos.gr</uni:has_email>
  <uni:has_age>54</uni:has_age>
  <uni:has_phone>2610000000</uni:has_phone>
  <uni:member_of>Paedagogical</uni:member_of>
  <uni:teaches>GreekI</uni:teaches>
</rdf:Description>

<rdf:Description rdf:about="http://www.mydomain.org/Student1">
  <rdf:type rdf:resource="uni:Student"/>
  <uni:has_name>Onoma1</uni:has_name>
  <uni:has_email>Onoma1@ceid.upatras.gr</uni:has_email>
  <uni:has_age>26</uni:has_age>
  <uni:has_phone>2610111111</uni:has_phone>
  <uni:member_of>CEID</uni:member_of>
</rdf:Description>
```

Για τη δημιουργία των μαθημάτων (Ενδεικτικό screenshot)

Όπως φαίνεται δεν γίνεται χρήση του **rdf:type** για να μην φαίνεται σε ποια κλάση ανήκει.

```
<rdf:Description rdf:about="http://www.mydomain.org/Web">
  <uni:les_name>Web development</uni:les_name>
  <uni:taught_by>Professor1</uni:taught_by>
</rdf:Description>
<rdf:Description rdf:about="http://www.mydomain.org/SoftwareQuality">
  <uni:les_name>Software Quality</uni:les_name>
  <uni:taught_by>Professor1</uni:taught_by>
</rdf:Description>
```

Για τη δημιουργία των αιθουσών (Ενδεικτικό screenshot)

3 για κάθε τμήμα συνολικά φτιάχτηκαν 18 classrooms

```
<rdf:Description rdf:about="http://www.mydomain.org/Classroom3">
  <rdf:type rdf:resource="uni:Classroom"/>
  <uni:room_name>D2</uni:room_name>
  <uni:room_capacity>100</uni:room_capacity>
  <uni:room_department>CEID</uni:room_department>
</rdf:Description>
<rdf:Description rdf:about="http://www.mydomain.org/Classroom4">
  <rdf:type rdf:resource="uni:Classroom"/>
  <uni:room_name>A1</uni:room_name>
  <uni:room_capacity>250</uni:room_capacity>
  <uni:room_department>Chemical Engineering</uni:room_department>
</rdf:Description>
```

## Ερώτημα 4

## Έγινε χρήση του ARQ JenaAPI

```
C:\Users\evans\OneDrive\Desktop\Αναπαράσταση γνώσης\2η άσκηση\apache-jena-4.3.1\bat>arq --data=test.rdf --query=test.rq
```

PREFIXES στο RQ αρχείο.

```
PREFIX uni: <http://www.mydomain.org/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/22-rdf-schema#>
```

- i) Αριστερά φαίνεται το query και δεξιά τι εμφανίζει

```
## 1ο ερώτημα ##
SELECT ?phone
WHERE {
    ?x rdf:type <uni:Professor> .
    ?x uni:has_phone ?phone .
}
```

```
phone
"26100000000"
"26100000000"
"26100000004"
"26100000000"
"26100000000"
"26100000000"
"26100000002"
"26100000000"
"26100000000"
"26100000001"
```

- ii) Για τους Students με ηλικία μεγαλύτερη του 23 έγινε χρήση Filter. Όπως φαίνεται για κάποιο λόγο το 23 πρέπει να μπει σε string μορφή αλλά εξακολουθεί να λειτουργεί σωστά. Αν αφαιρέσω τα “” παρατηρήθηκε ότι δεν επιστρέφει τίποτα.

```
#***** 2ο ερώτημα *****
SELECT ?phone
WHERE {
    ?student rdf:type <uni:Student>.
    ?student uni:has_age ?age.
    FILTER (?age>'23').
    ?student uni:has_phone ?phone.
}
```

[illegible]



- iii) Η κλάση Person είναι ουσιαστικά οι Students και οι Professors. Οπότε χρησιμοποιούμε union για να πάρουμε και τα 2 types. Στη συνέχεια εκμεταλλευόμαστε ότι έχουν και οι δυο κλάσεις properties με ίδια ονόματα και παίρνουμε το όνομά τους(property <uni:has\_name>) και το department που είναι καταχωρημένοι από την ιδιότητα <uni:member\_of>. Συγκρίνουμε αυτό με την ιδιότητα της κλάσης Department <uni:dep\_name> και παίρνουμε την ιδιότητα <uni:dep\_city> και βάζουμε ανάλογο filter.

```
##### 3ο ερώτημα #####
SELECT ?fullname
WHERE {
  {?name rdf:type <uni:Student>} UNION {?name rdf:type <uni:Professor>}.
  ?name uni:has_name ?fullname.
  ?name uni:member_of ?dep.
  ?c uni:dep_name ?dep;
    uni:dep_city ?city.
  FILTER(?city='Patra') .
}
```

fullname
"Onoma9"
"Onoma11"
"Onoma10"
"Onoma3"
"Onoma4"
"Onoma1"
"Onoma2"
"Onoma13"
"Onoma8"
"Onoma12"
"Professor4"
"Professor5"
"Professor3"
"Professor2"
"Professor1"

- iv) Αρχικά προστέθηκε ένα νέο PREFIX όπως φαίνεται στο screenshot. Παίρνουμε όλες τις ιδιότητες που έχει το classroom(**room\_name** , **room\_department** και **room\_capacity**). Αρχικά θέλουμε το capacity να είναι μεγαλύτερο του 150 (χρήση filter). Παρατηρήθηκε ότι δεν λειτουργούσε σωστά βάζοντας σε string το 150 και έπρεπε να γίνει χρήση της εντολής **xsd:integer()** (για αυτό προστέθηκε και το νέο PREFIX). Στη συνέχεια συγκρίνουμε το property του classroom **room\_department** με το property του Department **dep\_name** να είναι ίδια και με FILTER επιλέγουμε όσα departments έχουν στο **dep\_city** property τους την Πάτρα.

```
PREFIX uni: <http://www.mydomain.org/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/22-rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

##### 4ο ερώτημα #####
SELECT ?classname ?capacity
WHERE {
  ?x rdf:type <uni:Classroom>.
  ?x uni:room_name ?classname.
  ?x uni:room_department ?dep.
  ?x uni:room_capacity ?capacity.
  FILTER (xsd:integer(?capacity) > 150) .
  ?d uni:dep_name ?dep;
    uni:dep_city ?city.
  FILTER(?city='Patra') .
}
```

classname	capacity
"AB"	"200"
"A1"	"250"

## Ερώτημα 5

Για τη δημιουργία της εφαρμογής χρησιμοποιήθηκε το Apache Netbeans IDE 12.6 με βιβλιοθήκες από το JENA API.

- A) Για το πρώτο ερώτημα τελικό screenshot από το UI. Όπως φαίνεται έχει επιλεγθεί από το dropdown menu πάνω δεξιά μια από τα Departments και οι πίνακες αριστερά δείχνουν τα ανάλογα πράγματα.

The screenshot shows a Java Swing window titled 'Professor Table Details' with a dropdown menu on the right set to 'Fine Arts'. The window contains four tables: 'Professor Table Details', 'Student table details', 'Classrooms details', and 'Lessons details'. The 'ADD NEW RESOURCE' and 'URI SEARCH' buttons are on the right side.

Name	email	Age	phone	Department	Teaching
Professor6	email6@asf...	50	2610000000	Fine Arts	Photographyl
Professor7	email7@asf...	50	2610000000	Fine Arts	Photographyll

Name	email	Age	phone	Department
Onoma17	Onoma17@asf...	21	2610111111	Fine Arts
Onoma16	Onoma16@asf...	18	2610111111	Fine Arts
Onoma14	Onoma14@asf...	19	2610111111	Fine Arts

Classroom Name	Capacity	Department
C1	100	Fine Arts
C2	70	Fine Arts
C3	50	Fine Arts

Lesson	Professor	Department
Photography I	Professor6	Fine Arts
Photography II	Professor7	Fine Arts

Σχολιασμός κώδικα ερωτήματος A: Αρχικά γίνεται φόρτωση του αρχείου RDF και ένας έλεγχος αν είναι valid

```
//////////////// VALIDATOR //////////////////////////////////////
Model data = FileManager.get().loadModel("erwthma3.rdf");
InfModel infmodel = ModelFactory.createRDFModel(data);
ValidityReport validity = infmodel.validate();
if (validity.isValid()){
    System.out.print("OK");
} else {
    System.out.print("Conflicts");
    for (Iterator i = validity.getReports(); i.hasNext();){
        System.out.print("-" + i.next());
    }
}
```

Για το γέμισμα των πινάκων γίνεται χρήση τεσσάρων query (SPARQL ερωτήματα). Ένα για να πάρουμε ότι πληροφορία έχουμε για τους Professor του επιλεγμένου Department , ένα για τους Student , ένα για τα Classroom και ένα για τα Lesson.

```
//////////QUERY TO FILL CLASSROOM TABLE ////////////
DefaultTableModel aModel2 = (DefaultTableModel) ClassTable.getModel();
String classroom_queryString;
classroom_queryString = "PREFIX uni: <http://www.mydomain.org/>"
+ "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>"
+ "PREFIX rdfs: <http://www.w3.org/2000/01/22-rdf-schema#>"
+ "SELECT ?name ?capacity ?department "
+ "WHERE {"
+ "?x rdf:type <uni:Classroom> ."
+ "?x uni:room_name ?name."
+ "?x uni:room_capacity ?capacity ."
+ "?x uni:room_department ?department."
+ "}";

//////////QUERY TO FILL PROFESOR TABLE ////////////
DefaultTableModel aModel = (DefaultTableModel) ProfTable.getModel();
String prof_queryString;
prof_queryString = "PREFIX uni: <http://www.mydomain.org/>"
+ "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>"
+ "PREFIX rdfs: <http://www.w3.org/2000/01/22-rdf-schema#>"
+ "SELECT ?name ?mail ?age ?phone ?member ?teach "
+ "WHERE {"
+ "?x rdf:type <uni:Professor> ."
+ "?x uni:has_name ?name."
+ "?x uni:has_email ?mail ."
+ "?x uni:has_age ?age."
+ "?x uni:has_phone ?phone."
+ "?x uni:member_of ?member."
+ "?x uni:teaches ?teach."
+ "}";

//////////QUERY TO FILL LESSONS TABLE ////////////
DefaultTableModel aModel3 = (DefaultTableModel) LessonTable.getModel();
String Lesson_queryString;
// LESSONS DON'T HAVE RDF:TYPE FROM QUESTION 3 SO WE START FROM PROFESSOR
Lesson_queryString = "PREFIX uni: <http://www.mydomain.org/>"
+ "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>"
+ "PREFIX rdfs: <http://www.w3.org/2000/01/22-rdf-schema#>"
+ "SELECT ?lesson ?prof ?depart "
+ "WHERE {"
+ "?x rdf:type <uni:Professor> ."
+ "?x uni:has_name ?prof."
+ "?x uni:member_of ?depart."
+ "?y uni:taught_by ?prof."
+ "?y uni:les_name ?lesson."
+ "}";

//////////QUERY TO FILL STUDENT TABLE ////////////
DefaultTableModel aModel1 = (DefaultTableModel) StudentTable.getModel();
String student_queryString;
student_queryString = "PREFIX uni: <http://www.mydomain.org/>"
+ "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>"
+ "PREFIX rdfs: <http://www.w3.org/2000/01/22-rdf-schema#>"
+ "SELECT ?name ?mail ?age ?phone ?member "
+ "WHERE {"
+ "?x rdf:type <uni:Student> ."
+ "?x uni:has_name ?name."
+ "?x uni:has_email ?mail ."
+ "?x uni:has_age ?age."
+ "?x uni:has_phone ?phone."
+ "?x uni:member_of ?member."
+ "}";
```

Απλά για τη προσκόμιση των Lesson ανάλογα με το Department επειδή τα μαθήματα δεν ανήκουν σε κάποιο rdf:type και δεν έχει ιδιότητα με το όνομα του Department, εκμεταλλευόμαστε την ιδιότητα uni:taugh\_by των μαθημάτων και τι συγκρίνουμε με την ιδιότητα uni:has\_name των Professors μιας και εκεί υπάρχει και το όνομα του Department.

- Για το γέμισμα του Dropdown menu δεν έβαλα fixed τιμές. Αντιθέτως έγινε χρήση SPARQL για να γεμίσει το menu με τα Department ονόματα.

```
////////// QUERY TO FILL DROP DOWN MENU WITH DEP NAMES ////////////
String dep_names_queryString;
dep_names_queryString = "PREFIX uni: <http://www.mydomain.org/>"
+ "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>"
+ "PREFIX rdfs: <http://www.w3.org/2000/01/22-rdf-schema#>"
+ "SELECT ?y "
+ "WHERE {"
+ "?x rdf:type <uni:Department>."
+ "?x uni:dep_name ?y."
+ "}";
```

Έτσι όταν ο χρήστης προσθέτει νέο Department εμφανίζετε και αυτό.

- Για την αλλαγή των πινάκων ανάλογα με την επιλογή του Department από το dropdown menu έγινε η χρήση της βιβλιοθήκης TableRowSorter και φτιάχτηκε μια νέα συνάρτηση filter(String query) η οποία δέχεται και ένα όρισμα. Το String query προσκομίζεται από το dropdown selected value και φιλτράρει τους πίνακες ανάλογα με το String query.

```
private void filter (String query){
    ///FILTER professor TABLE WITH PARAMETER DRODOWN MENU SELECTED VALUE
    DefaultTableModel tblMode_filter_prof = (DefaultTableModel)ProfTable.getModel();
    TableRowSorter<DefaultTableModel> tr=new TableRowSorter<DefaultTableModel>(tblMode_filter_prof);
    ProfTable.setRowSorter(tr);
    tr.setRowFilter(RowFilter.regexFilter(query));
    /// FILTER STUDENT TABLE WITH PARAMETER DRODOWN MENU SELECTED VALUE
    DefaultTableModel tblMode_filter_student = (DefaultTableModel)StudentTable.getModel();
    TableRowSorter<DefaultTableModel> tr1=new TableRowSorter<DefaultTableModel>(tblMode_filter_student);
    StudentTable.setRowSorter(tr1);
    tr1.setRowFilter(RowFilter.regexFilter(query));
    ////// FILTER CLASSROOM TABLE WITH PARAMETER DRODOWN MENU SELECTED VALUE
    DefaultTableModel tblMode_filter_classroom = (DefaultTableModel)ClassTable.getModel();
    TableRowSorter<DefaultTableModel> tr2=new TableRowSorter<DefaultTableModel>(tblMode_filter_classroom);
    ClassTable.setRowSorter(tr2);
    tr2.setRowFilter(RowFilter.regexFilter(query));
    /////// FILTER LESSONS TABLE WITH PARAMETER DRODOWN MENU SELECTED VALUE
    DefaultTableModel tblMode_filter_lesson = (DefaultTableModel)LessonTable.getModel();
    TableRowSorter<DefaultTableModel> tr3=new TableRowSorter<DefaultTableModel>(tblMode_filter_lesson);
    LessonTable.setRowSorter(tr3);
    tr3.setRowFilter(RowFilter.regexFilter(query));
}
```

1<sup>ο</sup> screenshot η συνάρτηση filter

2<sup>ο</sup> screenshot που καλείτε η συνάρτηση και η προσκόμιση της τιμής της επιλεγμένης επιλογής από το drop down menu

```
private void jComboBox1ActionPerformed(java.awt.event.ActionEvent evt) {
    //GET DROP DOWN MENU SELECTED VALUE TO STRING
    String selectedValue = jComboBox1.getSelectedItem().toString();
    //CALL FILTER FUNCTION
    String filter_prof = selectedValue;
    filter(filter_prof);
}
```

Για την υλοποίηση των επόμενων ερωτημάτων φτιάχτηκαν 2 νέες οθόνες που έχουμε πρόσβαση μέσω 2 κουμπιών.

```
private void add_new_btnActionPerformed(java.awt.event.ActionEvent evt) {
    Add_new_property n = new Add_new_property();
    n.setVisible(true);
    this.dispose();
}
```

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    ypoerwthmaG g = new ypoerwthmaG();
    g.setVisible(true);
    this.dispose();
}
```

## B) UI Ερωτήματος B

back

Add new professor	Add new student	Add new department	Add new lesson	Add new classroom
Name: <input type="text"/>	Name: <input type="text"/>	Name: <input type="text"/>	Teacher: <input type="text"/>	Room name: <input type="text"/>
e-mail: <input type="text"/>	e-mail: <input type="text"/>	City: <input type="text"/>	Lesson Name: <input type="text"/>	Capacity: <input type="text"/>
age: <input type="text"/>	age: <input type="text"/>			Department: <input type="text"/>
Phone: <input type="text"/>	Phone: <input type="text"/>			
Department: <input type="text"/>	Department: <input type="text"/>			
Teaching: <input type="text"/>				
Add professor	Add student	Add department	Add lesson	Add classroom

Ενδεικτικό screenshot από την προσθήκη νέου Resource Professor. Σε όλα χρησιμοποιήθηκε η ίδια λογική.

```
Model modela = ModelFactory.createDefaultModel();
modela.read(in, null);
String uni = "http://www.mydomain.org/";
Resource ProfessorClass = modela.createResource("uni:Professor");
Resource New_Professor = modela.createResource(uni+prof_name);
Property p = modela.createProperty(uni, "has_name");
Property p1 = modela.createProperty(uni, "has_email");
Property p2 = modela.createProperty(uni, "has_age");
Property p3 = modela.createProperty(uni, "has_phone");
Property p4 = modela.createProperty(uni, "member_of");
Property p5 = modela.createProperty(uni, "teaches");
New_Professor.addProperty(RDF.type, ProfessorClass);
New_Professor.addProperty(p, prof_name);
New_Professor.addProperty(p1, prof_mail);
New_Professor.addProperty(p2, prof_age);
New_Professor.addProperty(p3, prof_phone);
New_Professor.addProperty(p4, prof_dep);
New_Professor.addProperty(p5, prof_lesson);
modela.write(new FileOutputStream("erwthma3.rdf"), "");
//modela.write(System.out, "");
} catch (NoWriterForLangException e) {
System.out.println("Invalid format option selected!");
} catch (FileNotFoundException ex) {
Logger.getLogger(Add_new_property.class.getName()).log(Level.SEVERE, null, ex);
}
```

Αρχικά φτιάχνουμε το νέο Resource του καθηγητη που θα προσθεθεί και μια «κλάση» που ουσιαστικά και αυτό είναι Resource αλλά το χρειαζόμαστε για το `rdf:type`. Στη συνέχεια φτιάχνονται οι ιδιότητες που έχει η κλάση καθηγητης και προστέθοντε στο Resource `New_Professor`. Προσθήτε και μια ιδιότητα που είναι ο τύπος του Resource με τη χρήση της εντολής `RDF.type, ProfessorClass` όπου `ProfessorClass` είναι η κλάση-Resource που φτιάχτηκε αρχικά. Στις υπόλοιπες ιδιότητες γίνεται εισαγωγή του εκάθωστε text-field input. Τέλος για την εισαγωγή του Resource στο RDF αρχείο γίνεται με τη χρήση εντολής `modela.write(output, "")`

Δείγμα εισαγωγής νέων Resource.

back

Add new professor	Add new student	Add new department	Add new lesson	Add new classroom
Name : Kwstas	Name: Panagiwths	Name: Physics	Teacher: Kwstas	Room name: P1
e-mail : stas@physics.gr	e-mail : pan@physics.gr	City: Athens	Lesson Name: Physics1	Capacity: 150
age : 30	age : 25			Department: Physics
Phone : 6988888888	Phone : 6933333333			
Department: Physics	Department: Physics			
Teaching : Physics1				
Add professor	Add student	Add department	Add lesson	Add classroom

CEID

CEID

Chemical Engineering

Fine Arts

Philosophy

Physics

Math

Paedagogical

Professor Table Details

Name	email	Age	phone	Department	Teaching
Kwstas	kwstas@ph...	30	6988888888	Physics	Physics1

Student table details

Name	email	Age	phone	Department
Panagiwths	pan@physics.gr	25	6933333333	Physics

Classrooms details

Classroom Name	Capacity	Department
P1	150	Physics

Lessons details

Lesson	Professor	Department
Physics1	Kwstas	Physics

Physics

ADD NEW RESOURCE

URI SEARCH

Και μέσα στο RDF αρχείο ενδεικτικά screenshots:

```
<rdf:Description rdf:about="http://www.mydomain.org/Physics1">
  <uni:taught_by>Kwstas</uni:taught_by>
  <uni:les_name>Physics1</uni:les_name>
</rdf:Description>
<rdf:Description rdf:about="http://www.mydomain.org/Kwstas">
  <rdf:type rdf:resource="uni:Professor"/>
  <uni:has_name>Kwstas</uni:has_name>
  <uni:has_email>kwstas@physics.gr</uni:has_email>
  <uni:has_age>30</uni:has_age>
  <uni:has_phone>6988888888</uni:has_phone>
  <uni:member_of>Physics</uni:member_of>
  <uni:teaches>Physics1</uni:teaches>
</rdf:Description>
```



Παρατηρήθηκε βέβαια κάτι που δεν φαίνεται να επηρεάζει κάπου τη λειτουργικότητα της εφαρμογής. Οι κλάσεις και τα properties στην αρχή του αρχείου που φτιάχτηκαν στο ερώτημα 3 αλλάζουν και παίρνουν μορφή Resources όπως φαίνεται παρακάτω. Αυτό γίνεται όταν προστίθεται νέο Resource μέσω της εφαρμογής.

```
<rdf:Description rdf:about="http://www.mydomain.org/has_name">
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:domain rdf:resource="http://www.mydomain.org/Person"/>
  <rdfs:range rdf:resource="https://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Description>

<rdf:Description rdf:about="http://www.mydomain.org/Student">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdfs:subClassOf rdf:resource="http://www.mydomain.org/Person"/>
</rdf:Description>
```

Γ)

Back

Enter a URI :

Search

Subject	Predicate	Object
http://www.mydomain.org/Kwstas	http://www.mydomain.org/teaches	Physics1
http://www.mydomain.org/Kwstas	http://www.mydomain.org/member_of	Physics
http://www.mydomain.org/Kwstas	http://www.mydomain.org/has_phone	6988888888
http://www.mydomain.org/Kwstas	http://www.mydomain.org/has_age	30
http://www.mydomain.org/Kwstas	http://www.mydomain.org/has_email	kwstas@physics.gr
http://www.mydomain.org/Kwstas	http://www.mydomain.org/has_name	Kwstas
http://www.mydomain.org/Kwstas	http://www.w3.org/1999/02/22-rdf-syntax-ns#...	uni:Professor
http://www.mydomain.org/Kwstas	http://www.w3.org/1999/02/22-rdf-syntax-ns#...	http://www.mydomain.org/Person
http://www.mydomain.org/Kwstas	http://www.w3.org/1999/02/22-rdf-syntax-ns#...	http://www.mydomain.org/Professor
http://www.mydomain.org/Kwstas	http://www.w3.org/1999/02/22-rdf-syntax-ns#...	http://www.w3.org/2000/01/rdf-schema#Res...

Για την προσκόμιση όλων των στοιχείων ανάλογα με το uri

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    String uri_txt = jTextField1.getText().toString();
    DefaultTableModel aModel = (DefaultTableModel) jTable1.getModel();
    aModel.setRowCount(0); // first clear existing table contents
    // get everything for selected URI
    Model data = FileManager.get().loadModel("erwthma3.rdf");
    InfModel infmodel = ModelFactory.createRDFSModel(data);
    Resource r = infmodel.getResource(uri_txt);
    StmtIterator iter = r.listProperties();
    while (iter.hasNext()) {
        Statement stmt = iter.nextStatement();
        System.out.print(" " + stmt.getSubject().toString());
        System.out.print(" " + stmt.getPredicate().toString());
        System.out.println(" " + stmt.getObject().toString());
        aModel.addRow(new Object[] {stmt.getSubject().toString(), stmt.getPredicate().toString(), stmt.getObject().toString()});
    }
    jTable1.setModel(aModel);
}
```

Για να καταφέρω να φαίνεται και ότι ένας Student ή Professor είναι και Person χωρίς αλλάξω το RDF γίνεται χρήση μοντέλου Συμπερασμού-Inference

```
InfModel infmodel = ModelFactory.createRDFSModel(data);
```

Και ένα ενδεικτικό screenshot για τα μαθήματα.

Back

Enter a URI :

Subject	Predicate	Object
http://www.mydomain.org/Python	http://www.mydomain.org/taught...	Professor1
http://www.mydomain.org/Python	http://www.mydomain.org/les_na...	Python
http://www.mydomain.org/Python	http://www.w3.org/1999/02/22-rdf..	http://www.mydomain.org/Lesson
http://www.mydomain.org/Python	http://www.w3.org/1999/02/22-rdf..	http://www.w3.org/2000/01/rdf-schema#Resource