

ΑΝΑΠΑΡΑΣΤΑΣΗ ΓΝΩΣΗΣ ΣΤΟΝ ΠΑΓΚΟΣΜΙΟ ΙΣΤΟ

1η ΣΕΙΡΑ ΑΣΚΗΣΕΩΝ

1 ΕΡΩΤΗΜΑ

Ως verifier χρησιμοποιείται το [XML](#) extension της Red Hat στο VSCode.

2 ΕΡΩΤΗΜΑ

Προσθέτουμε μέσα στο στοιχείο **Day** το τύπο `xsd:simpleType` με enumeration restrictions που προέρχονται βάσει του `xsd:string`. Τα enumerations έχουν τις τιμές όλων των ημερών της εβδομάδας.

Έτσι εμφανίζεται σφάλμα από τον verifier στο sample αρχείο `2_WRONG_schedule.xml`:

```
<Day>Tharsday</Day>
Value 'Tharsday' is not in the enumeration list.

It must be one of the following:
- Monday
- Tuesday
- Wednesday
- Thursday
- Friday
- Saturday
- Sunday

Code: xml(cvc-enumeration-valid)
cvc-type.3.1.3: The value 'Tharsday' of element 'Day' is not
```

3 ΕΡΩΤΗΜΑ

Δημιουργούμε την παρακάτω κανονική έκφραση:

```
[0-9]{2}:[0-9]{2}-[0-9]{2}:[0-9]{2}
```

Η μορφή `[0-9]{2}` αναπαριστά την ύπαρξη των δύο ψηφίων και έτσι εν τέλει περιγράφεται η μορφή `XX:XX-XX:XX`.

Ορίζουμε το τύπο του **Time** με παρόμοιο τρόπο όπως το δεύτερο ερώτημα μέσω του `SimpleType` και με restriction με βάση το `xsd:string`. Το restriction περιλαμβάνει ένα `xsd:pattern` που αποτελείται από την παραπάνω κανονική έκφραση.

Αν και είναι περιττό λόγω του regex, προσθέτουμε επιπλέον και το περιορισμό των 11 χαρακτήρων μέσω του `xsd:length`.

Έτσι πλέον εμφανίζεται σφάλμα στο `3_WRONG_schedule.xml`, μιας και το `09-10` δεν περιγράφεται από αυτή τη κανονική έκφραση:

```
cvc-pattern-valid: Value '9-11' is not facet-valid with respect to pattern '[0-9]{2}:[0-9]{2}-[0-9]{2}:[0-9]{2}' for type '#AnonType_TimeLectureEventSchedule'. xml(cvc-pattern-valid)
cvc-type.3.1.3: The value '9-11' of element 'Time' is not valid. xml(cvc-type.3.1.3)
View Problem No quick fixes available
<Time>9-11</Time>
```

Σημείωση: η εκφώνηση δεν προσδιορίζει συγκεκριμένα όρια στους ακεραίους, οπότε και τιμές όπως `54:80-75:90` θα γίνονται δεκτές.

4 ΕΡΩΤΗΜΑ

Προσθέτουμε στο element με όνομα **Lecture** τους περιορισμούς `minOccurs="1" maxOccurs="10"`. Επομένως στο σενάριο που υπήρχαν 11 lectures εμφανίζεται σφάλμα:

```
<
  cvc-complex-type.2.4.f: "'http://www.somesite.com/schema/schedule':Lecture' can
  occur a maximum of '10' times in the current sequence. This limit was exceeded.
  No child element is expected at this point. xml(cvc-complex-type.2.4.f)
  View Problem No quick fixes available
<Lecture>
  <Day>Wednesday</Day>
  <Time>09:00-10:00</Time>
```

5 ΕΡΩΤΗΜΑ

Στο τέλος του `xsd:sequence` του element **Lecture** προσθέτουμε το attribute με όνομα **Classroom** με τύπο `simpleType` που προέρχεται από το `xsd:string` με restriction να έχει μέχρι 8 χαρακτήρες. Η υποχρεωτικότητα επιτυγχάνεται μέσω του `use="required"`. Έτσι:

```
<
  Attribute 'Classroom' is missing from element 'Lecture'.
  Code: xml(cvc-complex-type.4)
  View Problem Quick Fix... (⌘.)
<Lecture>
  <Day>Thursday</Day>
```

```
p cvc-maxLength-valid: Value 'BAAAAAAA' with length = '8' is not facet-valid
t with respect to maxLength '7' for type
I '#AnonType_ClassroomLectureEventSchedule'. xml(cvc-maxLength-valid)
cvc-attribute.3: The value 'BAAAAAAA' of attribute 'Classroom' on element
' : 'Lecture' is not valid with respect to its type,
: '#AnonType_ClassroomLectureEventSchedule'. xml(cvc-attribute.3)
  View Problem No quick fixes available
  "BAAAAAAA">
```

6 ΕΡΩΤΗΜΑ

Μεταφέρουμε ξεχωριστά τα `complexType` και `simpleTypes` κάτω από το element **Schedule**.

7 ΕΡΩΤΗΜΑ

Μεταφέρουμε το `complexType` που περιλαμβάνει το element **Event** ξεχωριστά με όνομα `scheduleType`.

8 ΕΡΩΤΗΜΑ

lessonType: για την επέκταση χρησιμοποιούμε το `xsd:extension` και προσθέτουμε το element Professor.

seminarType: για τον περιορισμό χρησιμοποιούμε το `xsd:restriction` και θέτουμε το χαρακτηριστικό `maxOccurs="1"` στο Lecture.

labType: πάλι με το `xsd:restriction` αντιγράφουμε το eventType χωρίς το χαρακτηριστικό classroom.

9 ΕΡΩΤΗΜΑ

Δημιουργούμε τα παρακάτω στοιχεία τα οποία περνάμε ως `xsd:ref` στο scheduleType:

```
<xsd:element name="Lesson" substitutionGroup="Event" type="lessonType"/>
<xsd:element name="Seminar" substitutionGroup="Event" type="seminarType"/>
<xsd:element name="Lab" substitutionGroup="Event" type="labType"/>
```

Επειδή όμως το `schedule.xml` δεν περιλαμβάνει κάποια από τα στοιχεία που απαιτούνται από το ερώτημα 8, προκύπτουν errors (όπως για παράδειγμα την απουσία του Professor στο μάθημα "Constraint Satisfaction Problems").

10 ΕΡΩΤΗΜΑ

- `/Schedule/Lesson[Lecture[Day="Monday"]] //Title`

```
1. Constraint Satisfaction Problems
2. Artificial Intelligence
3. AI Programming
```

Με το `/Schedule/Lesson[Lecture[Day="Monday"]]` επιλέγουμε τα Lessons όπου έχουν ως δεύτερο παιδί τους το element Day να ισούται με Monday. Με το `//Title` επιλέγουμε όλα τα Titles. Η τομή τους δίνει τα Titles της Δευτέρας.

- `/Schedule/Lesson[Lecture[@Classroom="BA"]] //Title`

```
1. Artificial Intelligence
```

Με το `/Schedule/Lesson[Lecture[@Classroom="BA"]]` επιλέγονται όλα τα Lessons που έχουν σε παιδί το attribute Classroom=BA. Με παρόμοιο τρόπο επιλέγουμε την τομή τους μαζί με το `//Title`.

- `/Schedule/Lesson[Professor="Hatzilygeroudis"] //Lecture`

```
1. Wednesday 09-11
2. Thursday 09-11
3. Friday 15-17
4. Monday 11-13
```

Με παρόμοιο τρόπο επιλέγεται και η τομή του element Professor=Hatzilygeroudis μαζί με το Lecture

11 ΕΡΩΤΗΜΑ

Στην αρχή του αρχείου `schedule.xml` προστέθηκε η παρακάτω γραμμή ώστε να γίνει η ένωση με το αρχείο `schedulecatalog.xml`.

```
<?xml-stylesheet type="text/xsl" href="schedulecatalog.xml"?>
```

Στο αρχείο `schedulecatalog.xml` έχουμε δημιουργήσει ένα βασικό πρότυπο html όπου επικοινωνεί με το xml αρχείο για να δεχτεί τα δεδομένα του. Εκεί έχει περιληφθεί και το h2 header "Schedule". Για να ταξινομηθούν τα μαθήματα ανά ημέρα σε κάθε γραμμή του πίνακα χρησιμοποιούμε το όρισμα `xsl:for each` με την έκφραση σε XPath `Schedule/ Lesson[Lecture[Day='ΗΜΕΡΑ']]`. Στις τρεις στήλες τοποθετούνται οι τίτλοι και ο καθηγητής μέσω του `xsl:value of` και η ημέρα, η οποία είναι σταθερή κάθε φορά λόγω της `for`.

Το αποτέλεσμα είναι ο παρακάτω πίνακας:

Schedule

Title	Professor	Day
Constraint Satisfaction Problems		Monday
Artificial Intelligence		Monday
AI Programming	Hatzilygeroudis	Monday
Artificial Intelligence	Hatzilygeroudis	Wednesday
Introduction to Procedural Programming	Papadopoulos	Wednesday
Artificial Intelligence	Hatzilygeroudis	Thursday
Knowledge Representation in Web	Hatzilygeroudis	Friday