**ΑΝΑΠΑΡΑΣΤΑΣΗ ΓΝΩΣΗΣ ΣΤΟΝ ΠΑΓΚΟΣΜΙΟ ΙΣΤΟ**  
2020-2021  
**1η Σειρά Ασκήσεων**

**Ονοματεπώνυμο:** Αδαμόπουλος Κωνσταντίνος

**ΑΜ:** 6270 (1043750)

Σημείωση:

Για τα ερωτήματα από 1-11 χρησιμοποιήθηκε η σελίδα <https://www.freeformatter.com/> η οποία περιέχει όλα τα απαραίτητα εργαλεία για την διεκπεραίωση της άσκησης.

**Ερώτημα 1**

Screenshot για το validation του xml αρχείου που δόθηκε:



**Ερώτημα 2**

Άλλαξα τις τιμές που μπορούσε να πάρει το στοιχείο “Day” από οποιοδήποτε string τώρα μπορεί να πάρει μόνο μια μέρα της εβδομάδας(τύπου string).

**Ερώτημα 3**

Άλλαξα τις τιμές που μπορούσε να πάρει το στοιχείο “Time” ” από οποιοδήποτε string τώρα μπορεί να πάρει μόνο οποιαδήποτε τιμή που ακολουθεί το pattern ([0-1][0-9]|2[0-3]):([0-5][0-9])-([0-1][0-9]|2[0-3]):([0-5][0-9]) (τύπου string)..

**Ερώτημα 4**

Περιόρισα των αριθμό των εμφανίσεων του στοιχείου “Lecture” από αόριστο αριθμό σε ελάχιστο αριθμό εμφάνισης 1 και μέγιστο αριθμό εμφάνισης 9 όπως ζητείται στην εκφώνηση με την χρήση των **attributes** minOccurs, maxOccurs στα οποία τέθηκαν οι τιμές 1 και 9 αντίστοιχα.

**Ερώτημα 5**

Πρόσθεσα στο στοιχείο “Lecture” το attribute Classroom το οποίο δέχεται strings αλλά με μέγιστο μήκος χαρακτήρων μικρότερο του 8 οπότε maxLength=7.

**Ερώτημα 6A**

Αφαίρεσα τα στοιχεία που απαρτίζουν το “Event” στοιχείο από την inline δήλωση τους και τα τοποθέτησα σε ένα <xsl:complexType> tag με όνομα “eventType”, και στο στοιχείο “Event” ανάθεσα στο attribute type το “eventType” .

**Ερώτημα 6Β**

Αφαίρεσα τις inline δηλώσεις από όλα τα στοιχεία που απαρτίζουν το “Event” και ξανά δημιούργησα τους τύπους των στοιχείων και των χαρακτηριστικών(attributes) καθολικά και ανάθεσα στο attribute type των στοιχείων τον αντίστοιχο τύπο.

**Ερώτημα 7**

Αφαίρεσα τη δημιουργία του στοιχείου “Event” από το scope του στοιχείου “Schedule” και το δημιουργώ καθολικά. Μετά φτιάχνω ένα καινούργιο στοιχείο στο “Schedule” και στο attribute **ref** του στοιχείου εισάγω το όνομα του στοιχείου “Event”.

**Ερώτημα 8**

Προσθέτω τους επιπλέον τύπους οι οποίοι είναι οι: lessonType, seminarType, labType και είναι όλοι τύπου eventType και ο καθένας έχει μια διαφορετική προσθήκη ή περιορισμό .Στην περίπτωση του lessonType προστίθεται ένα επιπλέον στοιχείο με το όνομα “Professor”,στο seminarType ο αριθμός των διαλέξεων περιορίζεται σε 1 και στο labType απαγορεύεται η χρήση του attribute Classroom στα στοιχεία των “Lecture”.

**Ερώτημα 9**

Αφαίρεσα το use attribute apo to attribute Classroom στον τύπο “lecture\_t” διότι εφόσον υπάρχει η τιμή required στο use δεν μπορεί να αλλάξει σε άλλη τιμή στην περίπτωση μας σε prohibited. Πρόσθεσα τα στοιχεία “Lesson”, ”Seminar” και “Lab” και τα έθεσα σαν εναλλακτικά tags για το στοιχείο “Event” τα οποία ουσιαστικά αντικαθιστούν το στοιχείο “Event” αλλά και ακολουθούν τον τύπο τους.

**Screenshot για το Ερώτημα 2**

**Εικόνα που περιέχει κείμενο

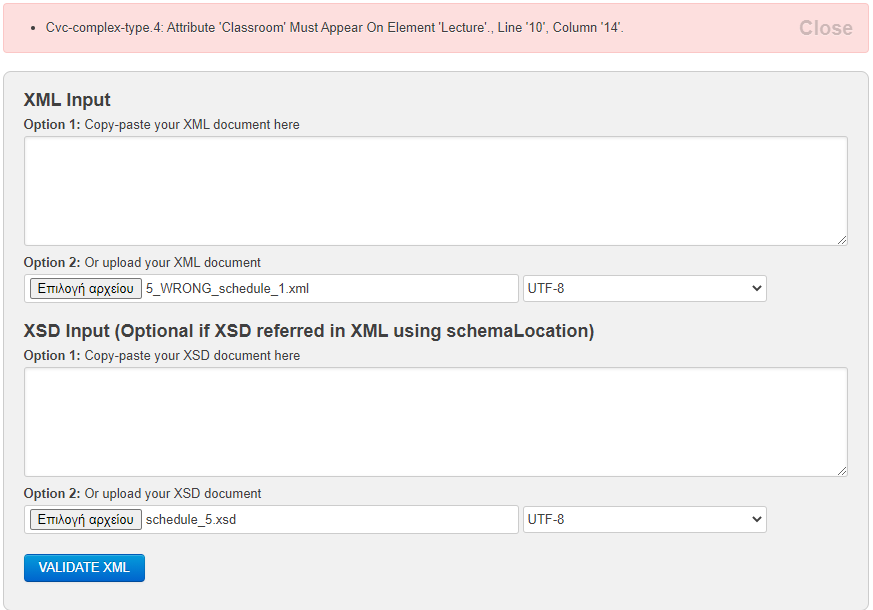
Περιγραφή που δημιουργήθηκε αυτόματα**

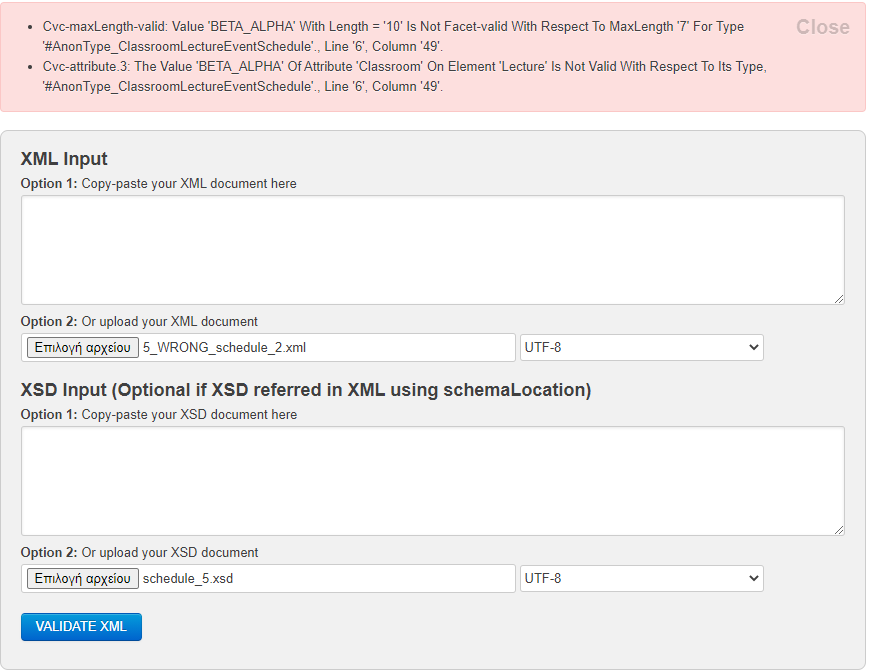
**Screenshot για το Ερώτημα 3**

**Εικόνα που περιέχει κείμενο

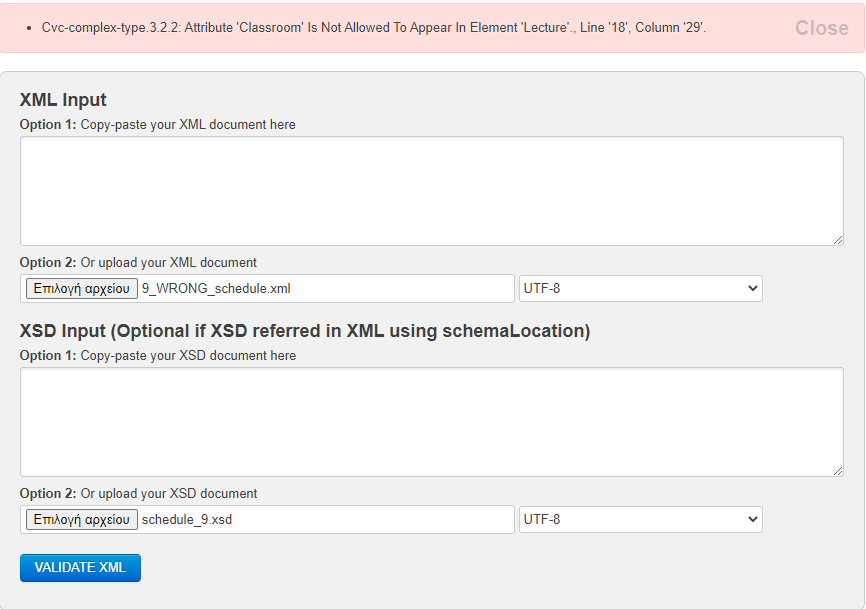
Περιγραφή που δημιουργήθηκε αυτόματα**

**Screenshots για το Ερώτημα 5**

****

****

**Screenshot για το Ερώτημα 9**

****

**Ερώτημα 10**

**Screenshots για τα αποτελέσματα των XPath queries**

Αρχείο δοκιμής των XPath queries: 10\_schedule.xml

* **XPATH: /Schedule/Lesson/Lecture[Day="Monday"]/../Title/text()**

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

* **XPATH: /Schedule/Lesson/Lecture[@Classroom="BA"]/../Title/text()**



* **XPATH: /Schedule/Lesson[Professor="Hatzilygeroudis"]/Lecture**

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

**Ερώτημα 11**

**Αρχείο μετασχηματισμού XSLT:**

Η αλλαγή που έγινε στο δοκιμαστικό αρχείο(10\_schedule.xml) είναι να προστεθεί η γραμμή:

1. **<?xml-stylesheet** type="text/xsl" href="transform.xsl" **?>**

Στο αρχείο .xslt εισάγω το **<xsl:template match=”/Schedule” >** το οποίο συσχετίζει το template με το στοιχείο που έχει εισαχθεί στο match attribute.Έπειτα δημιουργώ έναν πίνακα με headers τα Lesson, Professor, Day.Το ζητούμενο είναι να ταξινομήσουμε τα μαθήματα με βάση την ημέρα που διδάσκονται, για να γίνει αυτό εισάγω 5 for-each loops όσες και η μέρες(εργάσιμες) της εβδομάδας. Η κάθε loop τρέχει για όλα τα στοιχεία “Lecture” και εισάγει στον πίνακα μόνο τα μαθήματα που γίνονται την συγκεκριμένη ημέρα(διαφορετική για την κάθε loop) τον τίτλο του μαθήματος αλλά και τον καθηγητή που διδάσκει το μάθημα εφόσον έχει δηλωθεί και χρωματίζω με διαφορετικό χρώμα τις γραμμές που έχουν διαφορετικές ημέρες. Για να πάρω τις τιμές για τα πεδία του καθηγητή και του τίτλου χρησιμοποιώ τις **XPath εκφράσεις "../Professor" και "../Title"** οι οποίες μας πηγαίνουν στον πατέρα(“Lesson”) τους η καθεμία και έπειτα παίρνουν το στοιχείο Professor και Title αντίστοιχα τα οποία είναι παιδία του “Lesson”.

Παρακάτω παρουσιάζεται ο κώδικας του xslt αρχείου

1. **<?xml** version="1.0"**?>**
2. **<xsl:stylesheet** version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"**>**
3. **<xsl:template** match="/Schedule"**>**
4. **<html>**
5. **<head>**
6. **<title>**Results**</title>**
7. **</head>**
8. **<body>**
9. **<table>**
10. **<tr** style="background-color:gray;"**>**
11. **<th>**Lesson**</th>**
12. **<th>**Professor**</th>**
13. **<th>**Day**</th>**
14. **</tr>**
15. **<xsl:for-each** select="//Lecture"**>**
16. **<xsl:choose>**
17. **<xsl:when** test="Day/text()='Monday'"**>**
18. **<tr** style="background-color:#D241D5"**>**
19. **<td><xsl:value-of** select="../Title"**/></td>**
20. **<td><xsl:value-of** select="../Professor"**/></td>**
21. **<td><xsl:value-of** select="Day"**/></td>**
22. **</tr>**
23. **</xsl:when>**
24. **</xsl:choose>**
25. **</xsl:for-each>**
26. **<xsl:for-each** select="//Lecture"**>**
27. **<xsl:choose>**
28. **<xsl:when** test="Day/text()='Tuesday'"**>**
29. **<tr** style="background-color:#D2BC36"**>**
30. **<td><xsl:value-of** select="../Title"**/></td>**
31. **<td><xsl:value-of** select="../Professor"**/></td>**
32. **<td><xsl:value-of** select="Day"**/></td>**
33. **</tr>**
34. **</xsl:when>**
35. **</xsl:choose>**
36. **</xsl:for-each>**
37. **<xsl:for-each** select="//Lecture"**>**
38. **<xsl:choose>**
39. **<xsl:when** test="Day/text()='Wednesday'"**>**
40. **<tr** style="background-color:#64BCAA"**>**
41. **<td><xsl:value-of** select="../Title"**/></td>**
42. **<td><xsl:value-of** select="../Professor"**/></td>**
43. **<td><xsl:value-of** select="Day"**/></td>**
44. **</tr>**
45. **</xsl:when>**
46. **</xsl:choose>**
47. **</xsl:for-each>**
48. **<xsl:for-each** select="//Lecture"**>**
49. **<xsl:choose>**
50. **<xsl:when** test="Day/text()='Thursday'"**>**
51. **<tr** style="background-color:#64BC50"**>**
52. **<td><xsl:value-of** select="../Title"**/></td>**
53. **<td><xsl:value-of** select="../Professor"**/></td>**
54. **<td><xsl:value-of** select="Day"**/></td>**
55. **</tr>**
56. **</xsl:when>**
57. **</xsl:choose>**
58. **</xsl:for-each>**
59. **<xsl:for-each** select="//Lecture"**>**
60. **<xsl:choose>**
61. **<xsl:when** test="Day/text()='Friday'"**>**
62. **<tr** style="background-color:#FE6550"**>**
63. **<td><xsl:value-of** select="../Title"**/></td>**
64. **<td><xsl:value-of** select="../Professor"**/></td>**
65. **<td><xsl:value-of** select="Day"**/></td>**
66. **</tr>**
67. **</xsl:when>**
68. **</xsl:choose>**
69. **</xsl:for-each>**
70. **</table>**
71. **</body>**
72. **</html>**
74. **</xsl:template>**
75. **</xsl:stylesheet>**

Έξοδος όταν ανοίγω το xml αρχείο στο browser:

Εικόνα που περιέχει πίνακας

Περιγραφή που δημιουργήθηκε αυτόματα

**Ερώτημα 12**

**Σημείωση: Στο τέλος του εγγράφου παρουσιάζεται ο κώδικας του αρχείου ask12.py**

Η εφαρμογή γράφτηκε σε python 3.6 και χρησιμοποιήθηκε το module lxml για το parsing και την δημιουργία εγγραφών στο xml αρχείο και το PYQT5 για την δημιουργία του GUI. Το GUI παράχθηκε χρησιμοποιώντας την εφαρμογή Qt Designer.

Για να εγκαταστήσετε τα παραπάνω modules εφόσον υπάρχει η python στο PATH σας γράψτε python -m pip install pyqt5 lxml

Όνομα αρχείου που περιέχει το GUI ask12\_ui.py .

Όνομα του αρχείου που περιέχει τα ζητούμενα του ερωτήματος 12 ask12.py .

Για το testing της εφαρμογής χρησιμοποιήθηκαν τα 9\_RIGHT\_schedule.xml, 9\_WRONG\_schedule.xml και το schedule\_9.xsd ως αρχεία εισόδου xml και xsd(schema) αντίστοιχα.

Δημιουργώ μια κλάση με το όνομα xml\_handler στην οποία «δημιουργώ» το GUI(self.ui.setupUI) και συνδέω το click event των κουμπιών με τις κατάλληλες μεθόδους(\_\_addxml, \_\_addxsd, \_\_show2table, \_\_add2xml) και δημιουργώ και τα headers του πίνακα που θα χρησιμοποιήσω για την απεικόνιση των δεδομένων από το xml αρχείο.

Η μέθοδος **\_\_addxml** ανοίγει έναν file browser ο οποίος δέχεται μόνο xml αρχεία και μόλις διαλέξω ένα αρχείο το αποθηκεύω στην μεταβλητή self.path\_xml .

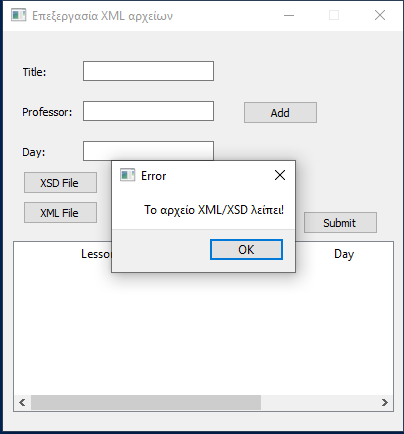
Η μέθοδος **\_\_addxsd** ανοίγει έναν file browser ο οποίος δέχεται μόνο xsd αρχεία και μόλις διαλέξω ένα αρχείο το αποθηκεύω στην μεταβλητή self.path\_xsd .

Στη μέθοδο **\_\_show2table**(εκτελείται όταν πατηθεί το κουμπί Υποβολή) αφού έχουμε εισάγει και το xml αρχείο και το xsd αρχείο κάνει parse το xml αρχείο και έπειτα ελέγχει για το αν είναι έγκυρο με βάση το xsd αρχείο που του δώσαμε. Αν όχι θα εμφανιστεί μήνυμα ότι το αρχείο δεν είναι έγκυρο, αλλιώς ότι το αρχείο είναι έγκυρο και έπειτα παίρνουμε την ρίζα του αρχείου και δύο for loops από τις οποίες η πρώτη βρίσκει τα παιδιά της ρίζας(‘Lesson’,’Lab’,’Seminar’) και η δεύτερη βρίσκει τα παιδία των παραπάνω παιδιών δηλαδή τα ‘Title’,’Lecture’ και ‘Professor’ αν υπάρχει. Επειδή στο αρχείο έχει δηλωθεί namespace όταν παίρνουμε το tag του στοιχείου μαζί με αυτό έρχεται και το namespace.Αυτό το λύνω με την εντολή etree.QName(elem).localname όπου elem ένα στοιχείο και η οποία μας επιστρέφει το τοπικό όνομα του στοιχείου. Έπειτα ελέγχω για το αν το στοιχείο έχει την τιμή ‘Title’ ή ‘Professor’ και αποθηκεύω τις text() τιμές τους στις προσωρινές μεταβλητές tmp\_title kai tmp\_prof. Στην περίπτωση του Professor tag κάνω την μεταβλητή flag ίσον με True που σημαίνει ότι έχει δηλωθεί καθηγητής και τον εισάγω τόσες φορές στην λίστα(profs) όσες και οι μέρες τις οποίες διδάσκει το μάθημα. Αμέσως μετά υπάρχει άλλη μία for loop η οποία υπάρχει ώστε να πάρουμε τις τιμές των στοιχείων Day(τις εισάγω στην λίστα days) η οποία καθορίζεται και από το ποια μέρα θέλει να πάρει αποτελέσματα ο χρήστης και ανάλογα πόσες μέρες διδάσκεται το μάθημα θα εισάγουμε το όνομα του μαθήματος στην λίστα titles και πριν προχωρήσω στο επόμενο ‘Lesson’ ,‘Lab’ ή ‘Seminar’ κοιτάω για το αν η μεταβλητή flag που αναφερθήκαμε παραπάνω έχει την τιμή False, αν ναι τότε εισάγω στην λίστα profs το “ ” τόσες φορές όσες και η μέρες που διδάσκεται το μάθημα το οποίο έχω βρει πιο πάνω και τέλος αφότου τελειώσει η επεξεργασία του αρχείου εισάγω τις τιμές των λιστών titles,profs και days στον πίνακα.

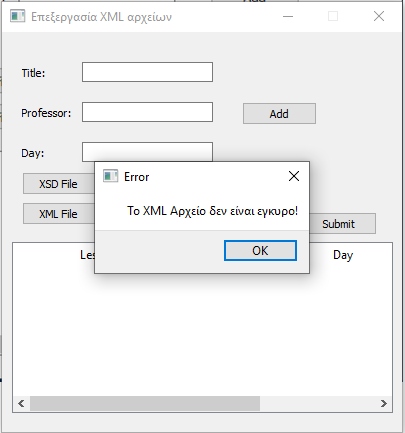
Στην μέθοδο **\_\_add2xml** ελέγχω για το αν έχει οριστεί το xml αρχείο, αν όχι τότε εμφανίζεται κατάλληλο μήνυμα αλλιώς κάνω parse το xml αρχείο παίρνω την ρίζα του αρχείου καθώς και τις τιμές που έχει εισάγει ο χρήστης στα πεδία ‘Title’ , ‘Professor’ και ‘Day’ ελέγχω αν τα πεδία ‘Title’ και ‘Day’ έχουν τιμές, αν όχι τότε εμφανίζεται σχετικό μήνυμα αλλιώς συνεχίζω και ελέγχω την τιμή που έχει το πεδίο ‘Day’. Αν δεν είναι ίσο με μια μέρα της εβδομάδας(εργάσιμη) τότε εμφανίζεται κατάλληλο μήνυμα αλλιώς προχωράμε στην δημιουργία των στοιχείων τα οποία θα περιέχουν τις τιμές που έχουμε ορίσει. Ξεκινάμε με την δημιουργία του στοιχείου ‘Lesson’ το οποίο είναι παιδί του root και το οποίο γίνεται με την εντολή etree.SubElement(root,'Lesson') με τον ίδιο τρόπο γίνεται και για το στοιχείο ‘Title’(και για όλα τα υπόλοιπα στοιχεία) , εισάγω την τιμή από από την μεταβλητή lesson στο στοιχείο ‘Title’ και μετά δημιουργώ το στοιχείο ‘Lecture’ το οποίο είναι παιδί του ‘Lesson’ και σε αυτό φτιάχνω άλλα δύο παιδία το ‘Day’ και το ‘Time’. Στο ‘Day’ εισάγεται η τιμή από την μεταβλητή day ενώ στο ‘Time’ εισάγεται μια τυχαία τιμή και αυτό γίνεται διότι το αρχείο το οποίο χρησιμοποιώ για να τεστάρω την εφαρμογή έχει αυτό το πεδίο. Έπειτα ελέγχω για το αν η μεταβλητή prof έχει τιμή αν ναι αυτό σημαίνει ότι έχει οριστεί τιμή στο πεδίο ‘Professor’ και δημιουργώ το κατάλληλο στοιχείο αλλιώς συνεχίζω στην εγγραφή/ανανέωση του αρχείου με τις νέες πληροφορίες.

Screenshots από την χρήση της εφαρμογής

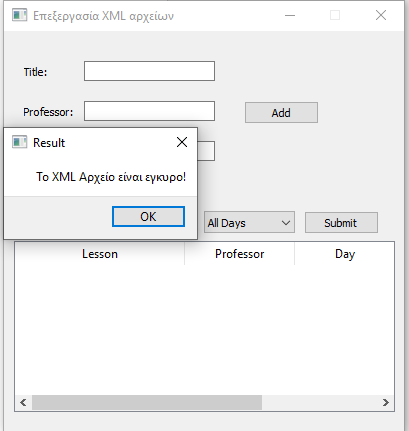
1. Αν δεν έχει εισαχθεί xml ή/και xsd αρχείο

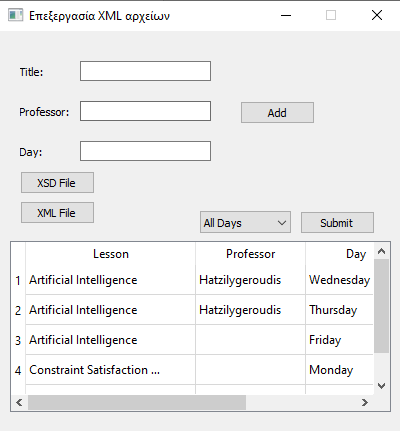


1. Αν έχει εισαχθεί λάθος xml αρχείο με βάση το xml schema(xsd αρχείο)

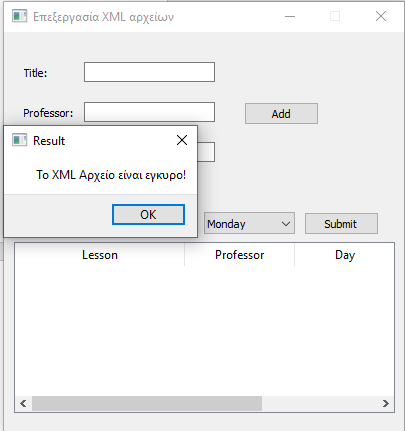


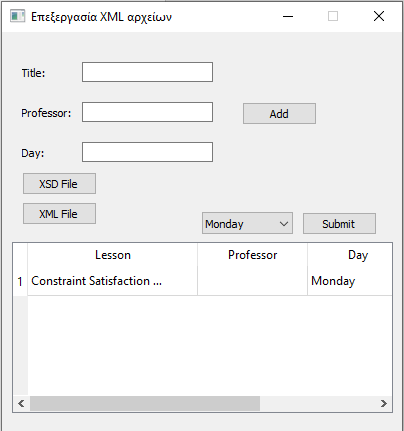
1. Αν εισάγω σωστό xml αρχείο με βάση το xml schema(xsd αρχείο) και θέσω σαν επιλογή για τις μέρες ‘All Days’



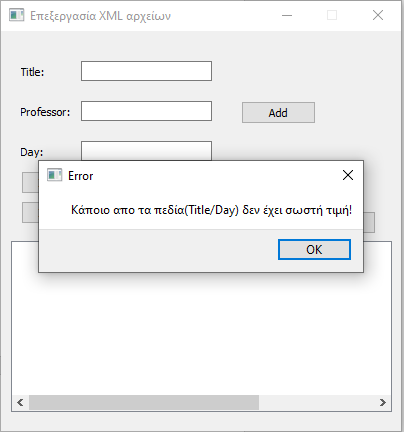


1. Αν εισάγω σωστό xml αρχείο με βάση το xml schema(xsd αρχείο) και θέσω σαν επιλογή για τις μέρες ‘Monday’

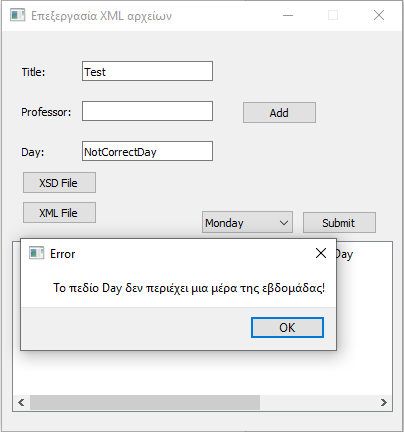




1. Αν τα πεδία ‘Title’ ή/και ‘Day’ δεν πάρουν τιμή



1. Αν το πεδίο ‘Day’ δεν πάρει την σωστή τιμή



1. Εισαγωγή σωστών τιμών στο πεδίο ‘Day’

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Τα περιεχόμενα του αρχείου μετά την εισαγωγή των παραπάνω στοιχείων

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

**Κώδικας του αρχείου ask12.py**

1. **from** PyQt5 **import** QtCore, QtGui, QtWidgets
2. **import** sys
3. **from** lxml **import** etree
5. **from** ask12\_ui **import** Ui\_XML\_Win
7. **class** xml\_handler(QtWidgets.QMainWindow):
8. **def** \_\_init\_\_(self):
9. super(xml\_handler,self).\_\_init\_\_()
10. self.ui = Ui\_XML\_Win()
11. self.ui.setupUi(self)
12. self.error\_dialog = QtWidgets.QMessageBox()
13. self.rows = 0 # αριθμος γραμμών του πινακα
15. # buttons
16. self.ui.xml\_input.clicked.connect(self.\_\_addxml)
17. self.ui.xsd\_input.clicked.connect(self.\_\_addxsd)
18. self.ui.submit.clicked.connect(self.\_\_show2table)
19. self.ui.add\_to\_xml.clicked.connect(self.\_\_add2xml)
21. #Table
22. self.model = QtGui.QStandardItemModel()
23. self.model.setHorizontalHeaderLabels(['Lesson','Professor','Day'])
24. self.ui.Data.setModel(self.model)
25. self.ui.Data.setColumnWidth(0,170)
26. self.ui.Data.setColumnWidth(1,110)
28. self.path\_xml = None # μονοπάτι του xml αρχείου
29. self.path\_xsd = None # μονοπάτι του xsd αρχείου
31. **def** \_\_addxml(self):
32. options = QtWidgets.QFileDialog.Options()
33. options |= QtWidgets.QFileDialog.DontUseNativeDialog
34. fileName, \_ = QtWidgets.QFileDialog.getOpenFileName(self,"Open", "","XML File (\*.xml)", options=options)
35. self.path\_xml = fileName

38. **def** \_\_addxsd(self):
39. options = QtWidgets.QFileDialog.Options()
40. options |= QtWidgets.QFileDialog.DontUseNativeDialog
41. fileName, \_ = QtWidgets.QFileDialog.getOpenFileName(self,"Open", "","XSD File (\*.xsd)", options=options)
42. self.path\_xsd = fileName
44. **def** \_\_show2table(self):
45. # καθαρισμος πινακα απο προηγούμενες εγγραφες
46. self.rows = 0
47. self.model.clear()
48. self.model.setHorizontalHeaderLabels(['Lesson','Professor','Day'])
49. self.ui.Data.setColumnWidth(0,170)
50. self.ui.Data.setColumnWidth(1,110)
51. xml\_f = None
52. xsd\_f = None
53. tmp\_prof = "" # προσωρινη μεταβλητη για τον καθηγητη
54. tmp\_title = "" # προσωρινη μεταβλητη για τον τιτλο
55. titles = [] # λιστα που περιεχει ολα τα μαθήματα(τιτλους)
56. profs = [] # λιστα που περιεχει ολους τους καθηγητες
57. days = [] # # λιστα που περιεχει ολες τις μερες που γινονται τα μαθήματα
58. flag = False # σημαία για το αν υπάρχει το στοιχείο Professor στο καθε Lesson αν οχι τοτε μπαινει κενο
59. day\_count = 0 # αριθμος μερων που γινεται το μαθημα
60. days\_filter = []
61. text = str(self.ui.filter.currentText())
62. **if**(text=="All Days"):
63. days\_filter = ["Monday","Tuesday","Wednesday","Thursday","Friday"]
64. **else**:
65. days\_filter.append(text)
66. **if**(self.path\_xsd **and** self.path\_xml):
67. xsd\_f = etree.parse(self.path\_xsd)
68. xsd = etree.XMLSchema(xsd\_f)
69. xml\_f = etree.parse(self.path\_xml)
70. **if**(xsd.validate(xml\_f)): # validation με το xsd αρχείο
71. self.error\_dialog.about(self,'Result','Το XML Αρχείο είναι εγκυρο!')
72. root = xml\_f.getroot()
73. **for** child **in** root.getchildren():
74. **for** elem **in** child.getchildren():
75. elem.tag = etree.QName(elem).localname # παιρνω το τοπικό ονομα και αποφευγω ετσι την εμφάνιση του namespace
76. **if** elem.tag=="Title":
77. tmp\_title = elem.text
78. **elif** elem.tag=="Professor":
79. tmp\_prof = elem.text
80. flag = True
81. **for** p **in** range(0,day\_count):
82. profs.append(tmp\_prof)
83. **for** day **in** elem.getchildren():
84. day.tag = etree.QName(day).localname
85. **if** day.tag=="Day" **and** day.text **in** days\_filter: # Αν βρει το στοιχείο Day και το text της ειναι ιδιο με την τιμη του φιλτρου
86. day\_count+=1
87. days.append(day.text)
88. titles.append(tmp\_title)
89. **if**(flag==False): # αν δεν υπαρχει Professor tag
90. **for** p **in** range(0,day\_count):
91. profs.append(" ")
92. day\_count=0
93. tmp\_prof = ""
94. flag = False
95. **for** title,prof,day **in** zip(titles,profs,days): # απεικονιση στοιχείων στον πινακα
96. qtitle = QtGui.QStandardItem(title)
97. qprof = QtGui.QStandardItem(prof)
98. qday = QtGui.QStandardItem(day)
99. self.model.setItem(self.rows,0,qtitle)
100. self.model.setItem(self.rows,1,qprof)
101. self.model.setItem(self.rows,2,qday)
102. self.rows+=1
104. **else**:
105. self.error\_dialog.about(self,'Error','Το XML Αρχείο δεν είναι εγκυρο!')
106. **return**
108. **else**:
109. self.error\_dialog.about(self,'Error','Το αρχείο XML/XSD λείπει!')
110. **return**
112. **def** \_\_add2xml(self):
113. **if**(**not** self.path\_xml):
114. self.error\_dialog.about(self,'Error','Το αρχείο XML λείπει!')
115. **return**
117. parser = etree.XMLParser(remove\_blank\_text=True)
118. xml\_f = etree.parse(self.path\_xml,parser)
119. root = xml\_f.getroot() # παιρνω την ριζα
120. lesson = self.ui.Lesson\_val.text() # τιμη του πεδίου Title
121. prof = self.ui.Professor\_val.text() # τιμη του πεδίου Professor
122. day = self.ui.Day\_val.text() # τιμη του πεδίου Day
123. **if**(**not** lesson **or** **not** day):
124. self.error\_dialog.about(self,'Error','Κάποιο απο τα πεδία(Title/Day) δεν έχει σωστή τιμή!')
125. **return**
126. **if** day **not** **in** ["Monday","Tuesday","Wednesday","Thursday","Friday"]:
127. self.error\_dialog.about(self,'Error','Το πεδίο Day δεν περιέχει μια μέρα της εβδομάδας!')
128. **return**
129. lesson\_elem = etree.SubElement(root,'Lesson')
130. title\_elem = etree.SubElement(lesson\_elem,'Title')
131. title\_elem.text = lesson
132. lect\_elem = etree.SubElement(lesson\_elem,'Lecture')
133. day\_elem = etree.SubElement(lect\_elem,'Day')
134. time\_elem = etree.SubElement(lect\_elem,'Time')
135. day\_elem.text = day
136. time\_elem.text = "10:30-12:30" # μια τυχαία ώρα
137. **if**(prof): # αν υπάρχει τιμη το πεδίο του Professor
138. prof\_elem = etree.SubElement(lesson\_elem,'Professor')
139. prof\_elem.text = prof
140. f = open(self.path\_xml,'w')
141. f.write(etree.tostring(root,pretty\_print=True).decode())
142. f.close()



147. **if** \_\_name\_\_=="\_\_main\_\_":
148. app = QtWidgets.QApplication(sys.argv)
149. window = xml\_handler()
150. window.show()
151. sys.exit(app.exec\_())