

ΑΝΑΠΑΡΑΣΤΑΣΗ ΓΝΩΣΗΣ ΣΤΟΝ ΠΑΓΚΟΣΜΙΟ ΙΣΤΟ

1η Σειρά Ασκήσεων

ΟΚΤΩΒΡΙΟΣ - ΝΟΕΜΒΡΙΟΣ 2022

Μάριος Στεφανίδης
1067458

An abstract graphic consisting of numerous thin, light purple lines that flow and curve across the bottom right portion of the cover, creating a sense of movement and depth.

Πίνακας Περιεχομένων

Ερώτημα 2	03
Ερώτημα 3	03
Ερώτημα 4	04
Ερώτημα 5	04
Ερώτημα 6	05
Ερώτημα 7	05
Ερώτημα 8	06
Ερώτημα 9	06
Ερώτημα 10	07
Ερώτημα 11	08
Ερώτημα 12	10



Εισήχθη ο καινούργιος τύπος δεδομένων

```
<xsd:simpleType name="specificDay">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Monday"/>
    <xsd:enumeration value="Tuesday"/>
    <xsd:enumeration value="Wednesday"/>
    <xsd:enumeration value="Thursday"/>
    <xsd:enumeration value="Friday"/>
    <xsd:enumeration value="Saturday"/>
    <xsd:enumeration value="Sunday"/>
  </xsd:restriction>
</xsd:simpleType>
```

```
<xsd:element name="Day" type="specificDay"/>
```

```
8      |<Day>Thurs</Day>  
❌ 2_WRONG_schedule.xml 1 of 2 problems  
  
Value 'Thursday' 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)
```

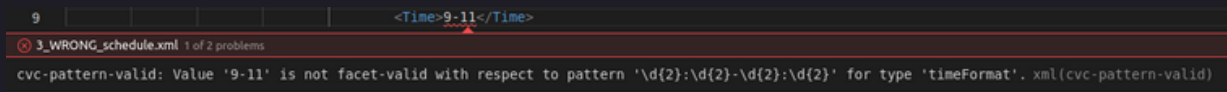
```
eFormat">  
="xsd:string">  
="11"/>  
e="\d{2}:\d{2}-\d{2}:\d{2}"/>
```

```
<xsd:simpleType name="timeFormat">
  <xsd:restriction base="xsd:string">
    <xsd:length value="11"/>
    <xsd:pattern value="\d{2}:\d{2}:\d{2}"/>
  </xsd:restriction>
</xsd:simpleType>
```

ενώ μεταβλήθηκε το στοιχείο Time ως εξής:

```
<xsd:element name="Day" type="specificDay"/>
```

Ο validator εντόπισε το εξής πρόβλημα στο αρχείο 3_WRONG_schedule:



```
9 | <Time>9-11</Time>
  | ^
  | 3_WRONG_schedule.xml 1 of 2 problems
  | cvc-pattern-valid: Value '9-11' is not facet-valid with respect to pattern '\d{2}:\d{2}-\d{2}:\d{2}' for type 'timeFormat'. xml(cvc-pattern-valid)
```

το οποίο είναι λογικό, αφού το 9-11 δεν ταιριάζει με το pattern που έχει οριστεί για το στοιχείο Time.

ΕΡΩΤΗΜΑ 4

Όπως φαίνεται στην παρακάτω εικόνα το στοιχείο Lecture μεταβλήθηκε ως εξής:

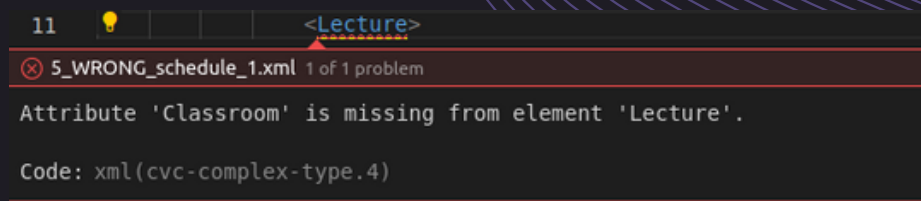
```
<xsd:element name="Lecture" minOccurs="1" maxOccurs="10">
```

ΕΡΩΤΗΜΑ 5

Προστέθηκε το attribute "Classroom" στο σημείο του schema που αφορά το στοιχείο Lecture ως εξής:

```
<xsd:attribute name="Classroom" use="required">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:minLength value="1"/>
      <xsd:maxLength value="8"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
```

Για τα δοκιμαστικά αρχεία που έχουν δοθεί, ο validator υπέδειξε τα παρακάτω λάθη:



5_WRONG_schedule_1

Το παραπάνω σφάλμα υποδεικνύει πως το στοιχείο Lecture πρέπει να χαρακτηρίζεται από κάποιο attribute με όνομα Classroom.



5_WRONG_schedule_2

Το παραπάνω σφάλμα υποδεικνύει πως η τιμή του attribute Classroom δεν μπορεί να ξεπερνάει τους 8 χαρακτήρες και όπως φαίνεται και στην παραπάνω εικόνα αυτός ο κανόνας παραβιάζεται.

ΕΡΩΤΗΜΑ 6

α) Σύμφωνα με την υπόδειξη που δίνεται στην εκφώνηση της άσκησης, οτιδήποτε είναι δηλωμένο ως "Event", τοποθετείται "έξω", προκειμένου να γίνει global ενώ ακόμα του δίνεται το όνομα "eventType". Τέλος, στο "Event" θέτουμε το type ίσο με "eventType".

β) Κατά τον ίδιο τρόπο υλοποιήθηκε και το συγκεκριμένο ερώτημα για τους τύπους "Lecture", "Day", "Time" και "Classroom".

ΕΡΩΤΗΜΑ 7

Προστίθεται το στοιχείο "Event" όπως φαίνεται και στην παρακάτω εικόνα:

```
<xsd:element name="Event" type="eventType"/>
```

ενώ το "Event" που βρίσκεται στο "Schedule" αντικαθίσταται ως εξής:

```
<xsd:element name="Schedule">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="Event" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

ΕΡΩΤΗΜΑ 8

Προστέθηκαν οι παρακάτω νέοι τύποι:

```
<xsd:complexType name="lessonType">
  <xsd:complexContent>
    <xsd:extension base="eventType">
      <xsd:sequence>
        <xsd:element name="Professor" type="xsd:string"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="seminarType">
  <xsd:complexContent>
    <xsd:restriction base="eventType">
      <xsd:sequence>
        <xsd:element name="Title" type="xsd:string"/>
        <xsd:element name="Lecture" type="lectureType" minOccurs="1" maxOccurs="10"/>
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="labType">
  <xsd:complexContent>
    <xsd:restriction base="eventType">
      <xsd:sequence>
        <xsd:element name="Title" type="xsd:string"/>
        <xsd:element name="Lecture" minOccurs="1" maxOccurs="10">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="Day"/>
              <xsd:element name="Time"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
```

ΕΡΩΤΗΜΑ 9

Προστέθηκαν οι παρακάτω νέοι τύποι:

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

Για το δοκιμαστικό αρχείο που έχει δοθεί, ο validator υπέδειξε το παρακάτω λάθος:

```
9_WRONG_schedule.xml 1 of 1 problem  
cvc-complex-type.3.2.2: Attribute 'Classroom' is not allowed to appear in element 'Lecture'. xml(cvc-complex-type.3.2.2)
```

Το παραπάνω σφάλμα είναι λογικό, αφού σύμφωνα με το καινούργιο restriction που δημιουργήθηκε μέσω του στοιχείου <labType>, το τελευταίο δεν μπορεί να περιέχει το χαρακτηριστικό classroom.

ΕΡΩΤΗΜΑ 10

Για την υλοποίηση των XPath ερωτημάτων χρησιμοποιήθηκε το site: [Free Online XPath Tester / Evaluator - FreeFormatter.com](https://www.freeformatter.com/xpath-tester.html). Παρακάτω αναφέρονται πρώτα τα ερωτήματα και έπειτα τα αποτελέσματα αυτών.

α) /Schedule/Lesson[Lecture[Day="Monday"]]/Title

Element='<Title>Constraint Satisfaction Problems</Title>'

Element='<Title>Artificial Intelligence</Title>'

Element='<Title>AI Programming</Title>'

β) /Schedule/Lesson[Lecture[@Classroom="BA"]]/Title

Element='<Title>Artificial Intelligence</Title>'

γ) //Lesson[Professor="Hatzilygeroudis"]/Lecture

Element='<Lecture Classroom="BA">

<Day>Wednesday</Day>

<Time>09-11</Time>

</Lecture>'

Element='<Lecture Classroom="BA">

<Day>Thursday</Day>

<Time>09-11</Time>

</Lecture>'

Element='<Lecture Classroom="P200">

<Day>Friday</Day>

<Time>15-17</Time>

</Lecture>'

```
Element='<Lecture Classroom="B3">
<Day>Monday</Day>
<Time>11-13</Time>
</Lecture>'
```

ΕΡΩΤΗΜΑ 11

Για την υλοποίηση του ερωτήματος χρησιμοποιήθηκε και αυτή τη φορά το site: [Free Online XPath Tester / Evaluator - FreeFormatter.com](#). Πιο συγκεκριμένα, αφού δημιουργήθηκε το xsl αρχείο δόθηκε μαζί με το xml αρχείο της 10ης άσκησης στο παραπάνω site, το οποίο στη συνέχεια δημιούργησε το ζητούμενο html αρχείο. Το αποτέλεσμα φαίνεται στην παρακάτω εικόνα:

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

Ο κώδικας του xsd αρχείου φαίνεται παρακάτω:

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <html>
      <body>
        <h2>Schedule</h2>
        <table border="1">
          <tr bgcolor="#949494">
            <th>Title</th>
            <th>Professor</th>
            <th>Day</th>
          </tr>
          <xsl:for-each select="Schedule/Lesson/Lecture">
            <xsl:sort select="Day"/>
            <tr>
              <xsl:choose>
                <xsl:when test="contains(./Day, 'Monday')">
                  <td bgcolor="#ff0000"><xsl:value-of select="../Title"/></td>
                  <td bgcolor="#ff0000"><xsl:value-of select="../Professor"/></td>
                  <td bgcolor="#ff0000"><xsl:value-of select="Day"/></td>
                </xsl:when>
                <xsl:when test="contains(./Day, 'Tuesday')">
                  <td bgcolor="#ffff00"><xsl:value-of select="../Title"/></td>
                  <td bgcolor="#ffff00"><xsl:value-of select="../Professor"/></td>
                  <td bgcolor="#ffff00"><xsl:value-of select="Day"/></td>
                </xsl:when>
                <xsl:when test="contains(./Day, 'Wednesday')">
                  <td bgcolor="#ff00ff"><xsl:value-of select="../Title"/></td>
                  <td bgcolor="#ff00ff"><xsl:value-of select="../Professor"/></td>
                  <td bgcolor="#ff00ff"><xsl:value-of select="Day"/></td>
                </xsl:when>
                <xsl:when test="contains(./Day, 'Thursday')">
                  <td bgcolor="#0000ff"><xsl:value-of select="../Title"/></td>
                  <td bgcolor="#0000ff"><xsl:value-of select="../Professor"/></td>
                  <td bgcolor="#0000ff"><xsl:value-of select="Day"/></td>
                </xsl:when>
                <xsl:when test="contains(./Day, 'Friday')">
                  <td bgcolor="#ffa500"><xsl:value-of select="../Title"/></td>
                  <td bgcolor="#ffa500"><xsl:value-of select="../Professor"/></td>
                  <td bgcolor="#ffa500"><xsl:value-of select="Day"/></td>
                </xsl:when>
                <xsl:otherwise>
                  <td><xsl:value-of select="../Title"/></td>
                  <td><xsl:value-of select="../Professor"/></td>
                  <td><xsl:value-of select="Day"/></td>
                </xsl:otherwise>
              </xsl:choose>
            </tr>
          </xsl:for-each>
        </table>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

Στον παραπάνω κώδικα όπως φαίνεται αρχικά έχουν δημιουργηθεί 3 στήλες σύμφωνα με την εκφώνηση (Title, Professor, Day). Έπειτα, δίνεται η διαδρομή που καταλήγει στο node Day (XSLT <for-each>), το οποίο συγκρίνεται κάθε φορά με τη βοήθεια της XSLT <choose>, προκειμένου να διαπιστωθεί με ποια μέρα της εβδομάδας ισούται. Ανάλογα με το when που "ενεργοποιείται" κάθε φορά, χρωματίζεται η κάθε γραμμή του πίνακα (σύμφωνα με την αντίστοιχη μέρα). Έχει χρησιμοποιηθεί και το <xsl:otherwise> κύριως για λόγους πληρότητας του κώδικα.

ΕΡΩΤΗΜΑ 12

Για τη δημιουργία της απλής εφαρμογής χρησιμοποιήθηκε το παρακάτω XML αρχείο:

```
<?xml version="1.0"?>
<Schedule>
  <Lesson>
    <Title>Artificial Intelligence</Title>
    <Lecture Classroom="BA">
      <Day>Wednesday</Day>
      <Time>09-11</Time>
    </Lecture>
    <Lecture Classroom="BA">
      <Day>Thursday</Day>
      <Time>09-11</Time>
    </Lecture>
    <Professor>Hatzilygeroudis</Professor>
  </Lesson>
  <Lesson>
    <Title>Constraint Satisfaction Problems</Title>
    <Lecture Classroom="B3">
      <Day>Monday</Day>
      <Time>19-21</Time>
    </Lecture>
    <Professor>Papadopoulos</Professor>
  </Lesson>
  <Lesson>
    <Title>Knowledge Representation in Web</Title>
    <Lecture Classroom="P200">
      <Day>Friday</Day>
      <Time>15-17</Time>
    </Lecture>
    <Professor>Hatzilygeroudis</Professor>
  </Lesson>
  <Lesson>
    <Title>Artificial Intelligence</Title>
    <Lecture>
      <Day>Monday</Day>
      <Time>19-21</Time>
    </Lecture>
    <Professor>Papadopoulos</Professor>
  </Lesson>
  <Lesson>
    <Title>AI Programming</Title>
    <Lecture Classroom="B3">
      <Day>Monday</Day>
      <Time>11-13</Time>
    </Lecture>
    <Professor>Hatzilygeroudis</Professor>
  </Lesson>
  <Lesson>
    <Title>Introduction to Procedural Programming</Title>
    <Lecture Classroom="P200">
      <Day>Wednesday</Day>
      <Time>15-17</Time>
    </Lecture>
    <Professor>Papadopoulos</Professor>
  </Lesson>
</Schedule>
```

Παρακάτω φαίνεται το GUI που έχει αναπτυχθεί για τη ζητούμενη εφαρμογή (χρησιμοποιήθηκε η Java και το περιβάλλον Netbeans για την υλοποίησή της) :

Lesson Title:

Professor:

Day: Add

Time:

Monday

Lesson Title	Professor	Day

show

Validate XML file

Result:

Επιλέγοντας το button show εμφανίζονται στον διπλανό πίνακα όλα τα μαθήματα που εμπεριέχονται στο αρχείο XML. Παρακάτω παρατίθενται ο κώδικας που εκτελείται (με αναλυτικά σχόλια), όταν επιλέγεται το παραπάνω button αλλά και τα περιεχόμενα του πίνακα:

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // Initialize jTable
    DefaultTableModel tblModel = (DefaultTableModel)jTable1.getModel();
    tblModel.setRowCount(0);

    final String FILENAME = "schedule.xml";

    // Instantiate the Factory
    DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();

    try {
        // parse XML file
        DocumentBuilder db = dbf.newDocumentBuilder();
        Document doc = db.parse(new File(FILENAME));

        // normalize
        doc.getDocumentElement().normalize();

        // get <Lesson>
        NodeList lessonList = doc.getElementsByTagName("Lesson");

        // iterate through the lessonList
        for (int i = 0; i < lessonList.getLength(); i++) {
            // get each <Lesson> node
            Node lessonNode = lessonList.item(i);

            if (lessonNode.getNodeType() == Node.ELEMENT_NODE) {
                // create the element
                Element Lesson = (Element) lessonNode;

                // get text of <Title> and <Professor>
                String Title = Lesson.getElementsByTagName("Title").item(0).getTextContent();
                String Professor = Lesson.getElementsByTagName("Professor").item(0).getTextContent();

                // get all the <Lecture> of the node <Lesson>
                NodeList lectureList = Lesson.getElementsByTagName("Lecture");

                // iterate through the lectureList
                for (int j = 0; j < lectureList.getLength(); j++) {
                    Node lectureNode = lectureList.item(j);

                    if (lectureNode.getNodeType() == Node.ELEMENT_NODE) {
                        Element Lecture = (Element) lectureNode;

                        String Day = Lecture.getElementsByTagName("Day").item(0).getTextContent();

                        String tblData[] = {Title, Professor, Day};
                        tblModel.addRow(tblData);
                    }
                }
            }
        }
    } catch (ParserConfigurationException | SAXException | IOException e) {}
}
```

Lesson Title	Professor	Day
Artificial Intelligence	Hatzilygeroudis	Wednesday
Artificial Intelligence	Hatzilygeroudis	Thursday
Constraint Satisfac...	Papadopoulos	Monday
Knowledge Repres...	Hatzilygeroudis	Friday
Artificial Intelligence	Papadopoulos	Monday

show

Μέσω του ComboBox ο χρήστης μπορεί να επιλέξει να εμφανιστούν στον πίνακα τα μαθήματα μιας συγκεκριμένης ημέρας. Παρακάτω φαίνονται τα αποτελέσματα του πίνακα για την ημέρα Δευτέρα (δεν παρατίθεται ο αντίστοιχος κώδικας, μιας και είναι πολύ παρόμοιος με την περίπτωση του button show με την διαφορά ότι έχει προστεθεί μια if):

Monday		
Lesson Title	Professor	Day
Constraint Satisfac...	Papadopoulos	Monday
Artificial Intelligence	Papadopoulos	Monday
AI Programming	Hatzilygeroudis	Monday

show

Συμπληρώνοντας τη φόρμα που υπάρχει στην εφαρμογή δίνεται η δυνατότητα στον χρήστη εισαγωγής δεδομένων στο αρχείο XML. Παρακάτω φαίνεται η συμπλήρωση της φόρμας, το αποτέλεσμα προσθήκης καινούργιου node στο αρχείο schedule.xml και ο κώδικας που γράφτηκε για το συγκεκριμένο ερώτημα (με αναλυτικά σχόλια):

Lesson Title:

Professor:

Day:

Time:

Add

Validate Schema

Result

Message

The node was appended successfully

OK

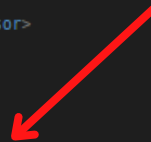
Όπως φαίνεται και στην εικόνα, εμφανίζεται μήνυμα επιτυχίας, το οποίο επιβεβαιώνει την προσθήκη του καινούργιου node.

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?><Schedule>
  <Lesson>
    <Title>Artificial Intelligence</Title>
    <Lecture Classroom="BA">
      <Day>Wednesday</Day>
      <Time>09-11</Time>
    </Lecture>
    <Lecture Classroom="BA">
      <Day>Thursday</Day>
      <Time>09-11</Time>
    </Lecture>
    <Professor>Hatzilygeroudis</Professor>
  </Lesson>
  <Lesson>
    <Title>Constraint Satisfaction Problems</Title>
    <Lecture Classroom="B3">
      <Day>Monday</Day>
      <Time>19-21</Time>
    </Lecture>
    <Professor>Papadopoulos</Professor>
  </Lesson>
  <Lesson>
    <Title>Knowledge Representation in Web</Title>
    <Lecture Classroom="P200">
      <Day>Friday</Day>
      <Time>15-17</Time>
    </Lecture>
    <Professor>Hatzilygeroudis</Professor>
  </Lesson>
  <Lesson>
    <Title>Artificial Intelligence</Title>
    <Lecture>
      <Day>Monday</Day>
      <Time>19-21</Time>
    </Lecture>
    <Professor>Papadopoulos</Professor>
  </Lesson>
  <Lesson>
    <Title>AI Programming</Title>
    <Lecture Classroom="B3">
      <Day>Monday</Day>
      <Time>11-13</Time>
    </Lecture>
    <Professor>Hatzilygeroudis</Professor>
  </Lesson>
  <Lesson>
    <Title>Introduction to Procedural Programming</Title>
    <Lecture Classroom="P200">
      <Day>Wednesday</Day>
      <Time>15-17</Time>
    </Lecture>
    <Professor>Papadopoulos</Professor>
  </Lesson>
  <Lesson>
    <Title>e-business</Title>
    <Lecture>
      <Day>Thursday</Day>
      <Time>9:00-11:00</Time>
    </Lecture>
    <Professor>Stefanidis</Professor>
  </Lesson>
</Schedule>

```

You, 1 second ago • Uncommitted changes



```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

    String Title = jTextField2.getText();
    String Professor = jTextField3.getText();
    String Day = jTextField4.getText();
    String Time = jTextField5.getText();

    if (Title.isEmpty() || Professor.isEmpty() || Day.isEmpty() || Time.isEmpty()) {

        JOptionPane.showMessageDialog(this, "You have to fill out all the whole form ");
        return;
    }

    File xmlFile = new File("schedule.xml");

    try {

        DocumentBuilderFactory docFactory = DocumentBuilderFactory.newInstance();
        DocumentBuilder docBuilder = docFactory.newDocumentBuilder();

        // root elements
        Document doc = docBuilder.parse(xmlFile);
        Element rootElement = doc.getDocumentElement();

        // add XML elements
        Element Lesson = doc.createElement("Lesson");
        // add <Lesson> to root
        rootElement.appendChild(Lesson);

        // create <Title> and its text
        Element elementTitle = doc.createElement("Title");
        elementTitle.setTextContent(Title);

        // append to node <Lesson>
        Lesson.appendChild(elementTitle);

        // create <Lecture> and its children
        Element Lecture = doc.createElement("Lecture");

        Element elementDay = doc.createElement("Day");
        elementDay.setTextContent(Day);

        Element elementTime = doc.createElement("Time");
        elementTime.setTextContent(Time);

        // append children to node <Lecture>
        Lecture.appendChild(elementDay);
        Lecture.appendChild(elementTime);

        // append <Lecture> to <Lesson>
        Lesson.appendChild(Lecture);

        // create <Professor> and its text
        Element elementProf = doc.createElement("Professor");
        elementProf.setTextContent(Professor);

        // append <Professor> to <Lesson>
        Lesson.appendChild(elementProf);

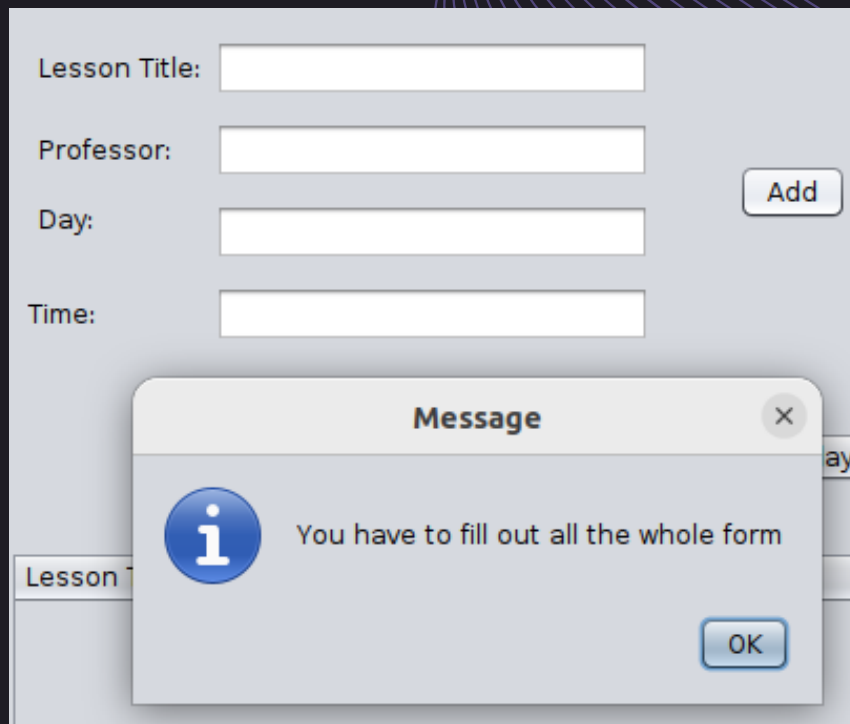
        Transformer tFormer = TransformerFactory.newInstance().newTransformer();
        tFormer.setOutputProperty(OutputKeys.INDENT, "xml");
        Source source = new DOMSource(doc);
        Result result = new StreamResult(xmlFile);
        tFormer.transform(source, result);

        JOptionPane.showMessageDialog(this, "The node was appended successfully");

    } catch (ParserConfigurationException | TransformerException | SAXException | IOException e) {}
}

```

Σε περίπτωση που κάποιο από τα πεδία δεν είναι συμπληρωμένο, τότε έχει υλοποιηθεί ένας απλός έλεγχος, ο οποίος εμφανίζει αντίστοιχο μήνυμα και διακόπτει τη λειτουργία της μεθόδου `jButtonActionPerformed`. Αυτό φαίνεται και στην παρακάτω εικόνα:



Επιλέγοντας το button Validate XML file, γίνεται validate το αρχείο schedule.xml μέσω του αντίστοιχου schema (schedule.xsd). Ανάλογως το αποτέλεσμα, ενημερώνεται το στοιχείο JLabel σε Result: OK! ή Result: NOT OK!. Παρατίθεται ο κώδικας που υλοποιεί το παραπάνω ερώτημα:

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    boolean value = validateXMLSchema("schedule.xsd", "schedule.xml");

    if (value) { jLabel6.setText("Result: OK!"); }
    else { jLabel6.setText("Result: NOT OK!"); }
}

private boolean validateXMLSchema(String xsdPath, String xmlPath) {
    try {
        SchemaFactory factory = SchemaFactory.newInstance(XMLConstants.W3C_XML_SCHEMA_NS_URI);
        Schema schema = factory.newSchema(new File(xsdPath));
        Validator validator = schema.newValidator();
        validator.validate(new StreamSource(new File(xmlPath)));
    }

    catch (IOException | SAXException e) {
        return false;
    }

    return true;
}
```

Σημείωση: Το περιβάλλον που χρησιμοποιήθηκε για την επεξεργασία και συγγραφή των XML και XSD αρχείων είναι το VS Code. Στο ερώτημα 12 παρατηρήθηκε πως ενώ το αρχείο schedule.xml γίνεται validate μέσω του παραπάνω περιβάλλοντος, ο κώδικας που έχει χρησιμοποιηθεί στο περιβάλλον Netbeans προκειμένου να πραγματοποιηθεί το validation, εμφανίζει το εξής **error: There is not a complete functional mapping between the particles**. Δεν κατάφερα να εντοπίσω το λόγο που συμβαίνει αυτό.