ΔΙΑΔΙΚΤΥΟ ΚΑΙ ΕΦΑΡΜΟΓΕΣ

Εαρινό εξάμηνο 2012

Άσκηση 2η : XML και XSL Transformations

Ευρίσκεται και στο site του μαθήματος

Προθεσμία παράδοσης σε γραπτή μορφή **18.05.12**

Δίδεται ο πηγαίος κώδικας του servlet **XMLTransformerAskhsh**.java, το οποίο 'μορφώνεται' με ένα μετασχηματισμό κατά την αρχικοποίησή του (init). Όταν κληθή, σύμφωνα με τις επιθυμίες του χρήστη ως προς την παρουσίαση (βλ. παρακάτω το index.html), μετασχηματίζει μέσω του CarPresentor2.xsl το Cars.xml (ευρίσκονται και τα δύο στο WEB-INF) εμφανίζοντας το τελευταίο στον browser σε μορφή πίνακα. Όπως δίνεται, το πρόγραμμα φτιάχνεται (από αρχείο) το xsltDoc σαν SAXSource, δηλ. σαν μία ακολουθία συμβάντων, η οποία 'μορφώνει' το myTransformer της Transformer, για το πώς θα μετασχηματίζει.

```
import java.io.*;
import javax.xml.transform.stream.*;
import javax.xml.transform.*;
import javax.xml.transform.sax.SAXSource;
import javax.xml.transform.stream.StreamSource;
import javax.xml.transform.stream.StreamResult;
import org.xml.sax.*;
// for servlet:
import javax.servlet.*;
import javax.servlet.http.*;

import org.w3c.dom.*;  // for DOM (Java DOM)
import javax.xml.parsers.*;

import javax.xml.transform.dom.*;// for transformations

public class XMLTransformerAskhsh extends HttpServlet {
	ServletContext ctx;
	String absPath;		//absolute path to our files - see below
	SAXSource xsltDoc; TransformerFactory tF;
	Transformer myTransformer;// will be built at init, will be used by doGet
	Document doc;

	public void init(ServletConfig config) throws UnavailableException {
		System.out.println("Init start");
		try {
			ctx = config.getServletContext();   // we will use the 'contex' below
			absPath = ctx.getRealPath("/WEB-INF/CarPresentor2.xsl");
			xsltDoc = new SAXSource(new InputSource(absPath));
			tF = TransformerFactory.newInstance();
			DocumentBuilderFactory fact = DocumentBuilderFactory.newInstance();
			// absolutely important, to understand the meaning of prefix 'xslt' !!!!
			fact.setNamespaceAware(true);
			DocumentBuilder builder = fact.newDocumentBuilder();
			doc = builder.parse(absPath);
			System.out.println("Name of document element is " +  doc.getDocumentElement().getNodeName());
		    } catch (Exception e) {e.printStackTrace(); }
		System.out.println("Init end");
	}
	private void changeDomByColor(Document doc, String color) {
		NodeList nl = doc.getElementsByTagName("h1");
		Attr a = doc.createAttribute("style");
		a.setValue("background-color: "+color);
		nl.item(0).getAttributes().setNamedItem(a);
	}

	public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
		System.out.println("dopost start");
```

```
		System.out.println("Name of document element (at the post) is " + doc.getDocumentElement().getNodeName());

		String color = request.getParameter("color");
		System.out.println("You selected the color "  + color);

	System.out.println(doc.getElementsByTagName("h1").item(0).getAttributes().getNamedItem("style").getNodeValue());

		changeDomByColor(doc, color);

	System.out.println(doc.getElementsByTagName("h1").item(0).getAttributes().getNamedItem("style").getNodeValue());

		PrintWriter pwr = response.getWriter();
		try {
			DOMSource ds = new DOMSource(doc) ;
			System.out.println( ((Document)ds.getNode()).getDocumentElement().getNodeName() +" "
+((Document)ds.getNode()).getDocumentElement().getNodeValue() ) ;
//			myTransformer = tF.newTransformer(new DOMSource(doc));
//			myTransformer = tF.newTransformer(xsltDoc);
			myTransformer = tF.newTransformer(ds);

			StreamSource xmlSource = new StreamSource(ctx.getResourceAsStream("/WEB-INF/Cars.xml"));
			System.out.println("Sending back the xml tranformed into html");
			response.setContentType("text/html"); //in order to put in http body
			myTransformer.transform(xmlSource, new StreamResult(pwr));
			pwr.println("The response sent back as a page!");
			pwr.close();
		}
		catch (Exception e)
		{
			e.printStackTrace();
		}
		System.out.println("dopost stop");
	}

}
```

1. Μελετήστε, εγκαταστήστε στον Tomcat και δοκιμάστε το παραπάνω servlet. Το πρόγραμμα είναι σε Java DOM (όλο το Java DOM API υπάρχει στις σημειώσεις) και όχι σε JDOM (για το οποίο οι σημειώσεις δίνουν παραδείγματα / κώδικα). Αν θέλετε παραπάνω λεπτομέρειες και αντίστοιχο κώδικα για Java DOM και για μετασχηματισμούς – Transforms δείτε τις δύο παρουσιάσεις στο DOM_Transforms από το site.

   Η σελίδα που διατίθεται στο χρήστη για να εκφράσει την επιθυμία του ως προς την παρουσίαση (μπορείτε να την εντάξετε σαν index.html) είναι η

```
<HTML><HEAD><TITLE>test</TITLE></HEAD><BODY>
<FORM ACTION='http://localhost:8080/TransformerDir/TransformerAsk' METHOD='POST'>
<STRONG>Please select:<br> </STRONG><PRE>
<INPUT TYPE='radio' NAME='color' VALUE='red'>RED<BR>
<INPUT TYPE='radio' NAME='color' VALUE='green'>GREEN<BR>
<INPUT TYPE='radio' NAME='color' VALUE='blue' CHECKED>BLUE<BR>
</PRE><INPUT TYPE='submit' VALUE='OK'>
</FORM>
</BODY></HTML>
```

2. Προεργασία / εξάσκηση εκτός Tomcat – δοκιμη μόνο με xalan: Έχοντας σαν βάση το CarPresentor2.xsl, δώστε άλλο δικό σας και γενικό xsl, το οποίο θα εμφανίζει σε πίνακα (html) **κάθε κείμενο με την δομή** του Cars.xml, αλλά με *οποιαδήποτε* ονόματα στοιχείων. Μελετήστε την δομή του Cars.xml και θεωρείστε, ότι ο τρόπος που γράφονται οι εγγραφές (vehicle, vehicle, vehicle, ...) σε ένα Πίνακα (vehicles) είναι αμετάβλητος, ενώ η ακριβής ονοματολογία vehicles – vehicle – πεδία του κάθε vehicle είναι αυθαίρετος. Επομένως χρειάζεται να γραφεί ένα νέο – γενικό – xsl και 2-3 'Cars.xml' (Airplanes.xml, Disks.xml, κλπ), από τα οποία το *ίδιο πρόγραμμα* παράγει έναν ωραίο πίνακα. **Στο νέο δικό σας γενικό xsl να μην χρησιμοποιείτε το στοιχείο xsl:for-each αλλά τα xsl-template και xsl:apply-templates**.

3. Έχοντας σαν βάση το **XMLTransformerAskhsh** φτιάξτε μία εφαρμογή web, η οποία εμπεριέχει μερικές (2-3) δομές δικές σας (η κάθε μία περίπου σαν το Cars.xml). Σε μία αρχική σελίδα, ο χρήστης θα καθορίζει (α) ποια από τις 2-3 δομές θέλει να δει και (β) ποια παραλλαγή παρουσίασης (άλλα χρώματα, άλλη διάταξη στην σελίδα, κλπ) προτιμά. Οι παραλλαγές θα είναι απλές, λίγες (2-3) και συγκεκριμένες. Η παρουσίαση θα γίνεται με τον μετασχηματισμό που αναπτύξατε παραπάνω στο 2 και με τις 2-3 παραλλαγές αυτού. Οι παραλλαγές παρουσίασης αντιστοιχούν σε μετατροπές αυτού του ενός μετασχηματισμού μέσα στην μνήμη. Οι μετατροπές στον μετασχηματισμό θα πρέπει να γίνονται προγραμματιστικά, αφού κατά την αρχικοποίηση θα έχετε εισάγει και μετά θα χρησιμοποιείτε το xsl όχι σαν αρχείο, ούτε σαν SAXSource αλλά σαν DOMSource, δηλ. σαν μία δομή DOM στην μνήμη. Σε καμία περίπτωση δεν πρέπει σε κάθε νέα επίσκεψη στο site να ξαναδιαβάζεται αρχείο .xsl (εξαιρετικά χρονοβόρο). Τελικά θα επιστρέφεται στον browser μία ωραία μορφοποιημένη σελίδα με το περιεχόμενο (α) και την παραλλαγή παρουσίασης (β) που επέλεξε ο χρήστης.

   ΠΑΡΑΔΟΤΕΑ. Η παράδοση σε 'γραπτή μορφή' σύμφωνα με την παραπάνω προθεσμία περιλαμβάνει τον κώδικα και τα σχετικά outputs. Μετά την παραπάνω παράδοση, θα πρέπει να επιδειχθεί ο κώδικας στο εργαστήριο στην **ημέρα (Πέμπτη η Παρασκευή) και ώρα (μεταξύ 12.45 και 14.30) πού επιθυμείτε και αφού έχετε παραδώσει την 'γραπτή μορφή'**.

Έπονται τα αρχεία στα οποία αναφέρονται τα πάνω προγράμματα (περίπου όπως στις σημειώσεις)

Cars.xml

```
<?xml version="1.0"?>
<!-- REFERENCE TO THE STYLESHEET: -->
<!-- not present here, is prorgammatically arranged -->
<vehicles>
  <vehicle year="1996" make="Land Rover" model="Discovery">
    <mileage>36500</mileage>
    <color>black</color>
    <price>$22100</price>
 </vehicle>
  <vehicle year="1998" make="Land Rover" model="Discovery">
    <mileage>3550</mileage>
    <color>yellow</color>
    <price>$31995</price>
  </vehicle>
  <vehicle year="1996" make="Chevrolet" model="Suburban 2500">
    <mileage>49300</mileage>
    <color>green</color>
    <price>$25995</price>
  </vehicle>
</vehicles>
```

CarPresentor2.xsl

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <html>
      <head>
        <title>Used Vehicles</title>
      </head>
          <body background="Money.jpg">
        <h1 style="background-color: #446600;
          color: #FFFFFF; font-size: 20pt; text-align: center;
          letter-spacing: 1.0em">Used Vehicles</h1>
        <table align="center" border="2">
          <tr>
            <th>Year</th>
            <th>Make</th>
            <th>Model</th>
            <th>Mileage</th>
            <th>Color</th>
            <th>Price</th>
          </tr>
<!-- This template has had so far only presentation elements,  -->
<!--  without any reference to the target xml document -->
<!--  We now refer to the  the target xml document -->
          <xsl:for-each  select="vehicles/vehicle">
                    <!-- order-by="+ price" has been removed above -->
            <tr>
              <td><xsl:value-of select="@year"/></td>
              <td><xsl:value-of select="@make"/></td>
              <td><xsl:value-of select="@model"/></td>
              <td><xsl:value-of select="mileage"/></td>
              <td><xsl:value-of select="color"/></td>
              <td><xsl:value-of select="price"/></td>
            </tr>
          </xsl:for-each>
        </table>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

Λεπτά Σημεία (μέσα στην XMLTransformerAskhsh)

1ον. Πως φτιάχνομε DOMSource;

ΟΧΙ απευθείας από αρχείο, όπως γίνεται στην περίπτωση της SAXSource,

- αλλά αφού πρώτα διαβάσομε το αρχείο και δημιουργήσομε Document doc (τέτοιο παράδειγμα έχει δοθεί, βλ και παρακάτω)
- και μετά με DOMSource ds = new DOMSource(doc)

Στα παραπάνω προσοχή να βάλετε τα αναγκαία import (φαίνονται στα παραδείγματα που δώσαμε).

2ον. namespace aware

Όταν ο builder (βλ. κάτω) επεξεργάζεται ένα αρχείο κειμένου xml και προσπαθεί να διαχωρίζει τα ονόματα των στοιχείων xml, δεν διακρίνει ότι πριν την ':' δεν υπονοείται μέρος του ονόματος αλλά το πρόθεμα ενός namespace (στην συγκεκριμένη περίπτωση το xsl:). Ο builder δεν είναι 'namespace aware', δηλ. πνίγεται το namespace, στο οποίο δηλώνεται ότι πρόκειται για xsl πέραν του ότι είναι xml !

Θεραπεία:

```
DocumentBuilderFactory fact = DocumentBuilderFactory.newInstance();
fact.setNamespaceAware(true); // PREPEI NA SHKVTHEI AYTH H FLAG !!!!!!
DocumentBuilder builder = fact.newDocumentBuilder();
doc = builder.parse(absPath);
```