

Διαδίκτυο και Εφαρμογές

2^η Εργαστηριακή Άσκηση

XML και XSL Transformations



**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ
ΠΟΛΥΤΕΧΝΕΙΟ**

8^ο Εξάμηνο
2017-2018

Λεούσης Σάββας
Α.Μ. 03114945

XML και XSL Transformations

1

1.a XML Parsing using JAVA DOM

Στην συγκεκριμένη άσκηση ζητήθηκε ένα πρόγραμμα Java περιδιάβασης μέσα σε DOM οποιουδήποτε XML, το οποίο τυπώνει κάτω από επικεφαλίδες με όνομα 'Level n' όλους τους κόμβους (Nodes) του αντίστοιχου επιπέδου n, δίνοντας όνομα και τιμή του κάθε κόμβου καθώς και τα attributes αυτού. Παρακάτω παρατίθεται ο κώδικας Java ο οποίος υλοποιεί το ζητούμενο πρόγραμμα, το οποίο δέχεται ως είσοδο στις παραμέτρους του το αρχείο XML προς περιδιάβαση:

```

1 import org.w3c.dom.*;
2 import javax.xml.parsers.*;
3 import java.util.*;
4 import java.io.PrintWriter;
5
6 public class DOMNavigator {
7     public static int maxDepth=1;
8     public static ArrayList<ArrayList<Node>> Nodes;
9     public static void main(String[] args) {
10         try {
11             PrintWriter output = new PrintWriter("output.txt");
12             DocumentBuilderFactory fact = DocumentBuilderFactory.newInstance();
13             DocumentBuilder builder = fact.newDocumentBuilder();
14             Document doc = builder.parse(args[0]);
15             Nodes = new ArrayList<ArrayList<Node>>();
16             Nodes.add(new ArrayList<Node>());
17             boolean newLevel;
18             orderTree(doc.getDocumentElement(),Nodes.get(0) , 1);
19             for (int i = 0; i < Nodes.size(); i++){
20                 newLevel = true;
21                 for (int j = 0 ; j < Nodes.get(i).size(); j++){
22                     Node element = Nodes.get(i).get(j);
23                     if (element.getNodeType() == Node.TEXT_NODE && element.
getNodeValue().replaceAll("\n","").replaceAll("\t","").replaceAll(" ","").
equals("")) {
24                         continue;
25                     }
26                     if (element.getNodeType() == Node.ELEMENT_NODE && element.
getFirstChild().getNodeType() == Node.TEXT_NODE && !element.getFirstChild().
getNodeValue().replaceAll("\n","").replaceAll("\t","").replaceAll(" ","").
equals("")) {
27                         newLevel = printLevel(newLevel,i,output);
28                         output.println("~~~ Node Name: " + element.getNodeName() +
" - Node Value: " + element.getFirstChild().getNodeValue().replaceAll("\n", "
").replaceAll("\t", " ").replaceAll(" ", " "));
29                     }
30                     else if(element.getNodeType() == Node.ELEMENT_NODE){
31                         newLevel = printLevel(newLevel,i,output);
32                         output.println("~~~ Node Name: " + element.getNodeName() +
" - Node Value: " + element.getNodeValue());
33                     }
34                     NamedNodeMap cl = element.getAttributes();
35                     if (cl != null && cl.getLength() != 0) {
36                         output.println("\t\tAttributes:");
37                         for (int k = 0; k < cl.getLength(); k++) {

```

```

38         Node node = cl.item(k);
39         output.println("\t\t\t"+node.getNodeName() + " = "
+ node.getNodeValue());
40     }
41 }
42 }
43 }
44     output.close();
45 } catch (Exception e) {
46     e.getMessage();
47 }
48 }
49 public static void orderTree(Node doc,ArrayList<Node> nodeList , int depth) {
50     if (doc == null) {
51         System.out.println("Nothing to print!");
52         return;
53     }
54     nodeList.add(doc);
55     NodeList nl = doc.getChildNodes();
56     if (nl != null && nl.getLength() != 0) {
57         if (maxDepth<depth+1){
58             maxDepth = depth+1;
59             Nodes.add(new ArrayList<Node>());
60         }
61         for (int i = 0; i < nl.getLength(); i++) {
62             Node node = nl.item(i);
63             orderTree(node, Nodes.get(depth),depth + 1);
64         }
65     }
66 }
67 public static boolean printLevel(boolean cond,int i,PrintWriter output){
68     if (cond){
69         output.println("\n\t##### LEVEL " + (i+1) + "
#####\n");
70         cond = false;
71     }
72     return cond;
73 }
74 }

```

Listing 1: Το πρόγραμμα DOMNavigator.java

Το παραπάνω πρόγραμμα μπορεί να δεχθεί ως είσοδο οποιοδήποτε αρχείο XML και ως έξοδο δημιουργεί ένα αρχείο output.txt το οποίο περιέχει τα αποτελέσματα της περιδιάβασης. Για την άσκηση αυτή χρησιμοποιήθηκε ως ενδεικτική είσοδος το παρακάτω αρχείο XML:

```

1 <?xml version = "1.0"?>
2 <Level_1 attr="0">
3     <Level_2A attr = "1">
4         <Level_3AA>text</Level_3AA>
5         <Level_3AB>text</Level_3AB>
6         <Level_3AC>text</Level_3AC>
7         <Level_3AD>text</Level_3AD>
8     </Level_2A>
9     <Level_2B attr = "2">
10         <Level_3BA>text</Level_3BA>
11         <Level_3BB>text</Level_3BB>
12         <Level_3BC>text</Level_3BC>
13         <Level_3BD>text</Level_3BD>

```

```
14     </Level_2B>
15     <Level_2C attr = "3" attr2 = "4">
16         <Level_3CA>text</Level_3CA>
17         <Level_3CB>text</Level_3CB>
18         <Level_3CC>text</Level_3CC>
19         <Level_3CD>text</Level_3CD>
20     </Level_2C>
21 </Level_1>
```

Listing 2: Ένα ενδεικτικό XML αρχείο

Δίνοντας λοιπόν ως είσοδο το παραπάνω XML αρχείο, το πρόγραμμα παράγει την παρακάτω έξοδο:

```
1
2 ##### LEVEL 1 #####
3
4 ~~~ Node Name: Level_1 - Node Value: null
5     Attributes:
6         attr = 0
7
8 ##### LEVEL 2 #####
9
10 ~~~ Node Name: Level_2A - Node Value: null
11     Attributes:
12         attr = 1
13 ~~~ Node Name: Level_2B - Node Value: null
14     Attributes:
15         attr = 2
16 ~~~ Node Name: Level_2C - Node Value: null
17     Attributes:
18         attr = 3
19         attr2 = 4
20
21 ##### LEVEL 3 #####
22
23 ~~~ Node Name: Level_3AA - Node Value: text
24 ~~~ Node Name: Level_3AB - Node Value: text
25 ~~~ Node Name: Level_3AC - Node Value: text
26 ~~~ Node Name: Level_3AD - Node Value: text
27 ~~~ Node Name: Level_3BA - Node Value: text
28 ~~~ Node Name: Level_3BB - Node Value: text
29 ~~~ Node Name: Level_3BC - Node Value: text
30 ~~~ Node Name: Level_3BD - Node Value: text
31 ~~~ Node Name: Level_3CA - Node Value: text
32 ~~~ Node Name: Level_3CB - Node Value: text
33 ~~~ Node Name: Level_3CC - Node Value: text
34 ~~~ Node Name: Level_3CD - Node Value: text
```

Listing 3: Η έξοδος του προγράμματος

1.b XML Parsing using SAX

Στην συγκεκριμένη άσκηση ζητήθηκε πρόγραμμα Java με την ίδια λειτουργικότητα και έξοδο με το πρόγραμμα του ερωτήματος (1α), αλλά με επεξεργασία των SAX Events. Παρακάτω παρατίθεται ο κώδικας Java ο οποίος υλοποιεί το ζητούμενο πρόγραμμα, το οποίο δέχεται ως είσοδο στις παραμέτρους του το αρχείο XML προς περιδιάβαση:

```

1 import org.xml.sax.*;
2 import org.xml.sax.helpers.*;
3 import javax.xml.parsers.*;
4 import java.util.ArrayList;
5 import java.io.PrintWriter;
6
7 public class XMLEventsPresentor implements ContentHandler
8 {
9     public static class Node {
10         public String nodeName;
11         public String nodeValue;
12         public Attributes attr;
13         public Node(String name, Attributes attributes){
14             nodeName = name;
15             attr= attributes;
16         }
17     }
18     public static ArrayList<ArrayList<Node>> nodeList;
19     public static boolean foundTitle = false;
20     public static int a=-1;
21     private Locator lc;
22     public static void main(String[] args) throws Exception
23     {
24         XMLEventsPresentor xmlep = new XMLEventsPresentor();
25         nodeList = new ArrayList<ArrayList<Node>>();
26         PrintWriter output = new PrintWriter("output.txt");
27         xmlep.go(args[0]);
28         for (int i = 0; i < nodeList.size(); i++){
29             output.println("\n\t##### LEVEL " + (i+1) + "
#####\n");
30             for (int j = 0 ; j < nodeList.get(i).size(); j++){
31                 Node element = nodeList.get(i).get(j);
32                 if (element.nodeValue.replaceAll("\n","").replaceAll("\t","").
replaceAll(" ", "").equals("")){
33                     output.println("~~~ Node Name: " + element.nodeName + " - Node
Value: null");
34                 }
35                 else {
36                     output.println("~~~ Node Name: " + element.nodeName + " - Node
Value: " + element.nodeValue.replaceAll("\n", "").replaceAll("\t", "").
replaceAll(" ", ""));
37                 }
38                 if (element.attr.getLength() > 0) {
39                     output.println("\t\tAttributes:");
40                     for (int k = 0; k < element.attr.getLength(); k++){
41                         output.println("\t\t\t"+element.attr.getLocalName(k) + " =
" + element.attr.getValue(k));
42                     }
43                 }
44             }
45         }
46         output.close();

```

```

47     }
48     public void go(String arg) throws Exception
49     {
50         SAXParserFactory spf = SAXParserFactory.newInstance();
51         SAXParser sp = spf.newSAXParser();
52         XMLReader sr = sp.getXMLReader();
53         sr.setContentHandler(this);
54         sr.parse(arg);
55     }
56     public void setDocumentLocator(Locator locator)
57     {
58         this.lc = locator ;
59     }
60     public void startDocument() throws SAXException {}
61     public void endDocument() throws SAXException {}
62     public void startPrefixMapping(String prefix, String uri) throws SAXException
63     {}
64     public void endPrefixMapping(String prefix) throws SAXException {}
65     public void startElement(String namespaceURI, String localName, String qName,
66     Attributes atts) throws SAXException
67     {
68         a++;
69         if (nodeList.size()-1 < a) {
70             nodeList.add(new ArrayList<Node>());
71         }
72         nodeList.get(a).add(new Node(qName, new AttributesImpl(atts)));
73         foundTitle = true;
74     }
75     public void endElement(String namespaceURI, String localName, String qName)
76     throws SAXException
77     {
78         a--;
79     }
80     public void characters(char ch[], int start, int lenght) throws SAXException {
81         if (foundTitle){
82             nodeList.get(a).get(nodeList.get(a).size()-1).nodeValue = new String(
83             ch,start,lenght);
84             foundTitle = false;
85         }
86     }
87     public void ignorableWhitespace(char ch[], int start, int lenght) throws
88     SAXException {}
89     public void processingInstruction(String target, String data) throws
90     SAXException {}
91     public void skippedEntity(String name) throws SAXException {}
92 }

```

Listing 4: Το πρόγραμμα XMLEventsPresenter.java

Το παραπάνω πρόγραμμα μπορεί να δεχθεί ως είσοδο οποιοδήποτε αρχείο XML και ως έξοδο δημιουργεί ένα αρχείο output.txt το οποίο περιέχει τα αποτελέσματα της περιδιάβασης. Για την άσκηση αυτή χρησιμοποιήθηκε ως ενδεικτική είσοδος το παρακάτω αρχείο XML:

```

1 <?xml version = "1.0"?>
2 <Level_1 attr="0">
3     <Level_2A attr = "1">
4         <Level_3AA>text</Level_3AA>
5         <Level_3AB>text</Level_3AB>
6         <Level_3AC>text</Level_3AC>

```

```

7      <Level_3AD>text</Level_3AD>
8  </Level_2A>
9  <Level_2B attr = "2">
10     <Level_3BA>text</Level_3BA>
11     <Level_3BB>text</Level_3BB>
12     <Level_3BC>text</Level_3BC>
13     <Level_3BD>text</Level_3BD>
14 </Level_2B>
15 <Level_2C attr = "3" attr2 = "4">
16     <Level_3CA>text</Level_3CA>
17     <Level_3CB>text</Level_3CB>
18     <Level_3CC>text</Level_3CC>
19     <Level_3CD>text</Level_3CD>
20 </Level_2C>
21 </Level_1>

```

Listing 5: Ένα ενδεικτικό XML αρχείο

Δίνοντας λοιπόν ως είσοδο το παραπάνω XML αρχείο, το πρόγραμμα παράγει την παρακάτω έξοδο:

```

1
2 ##### LEVEL 1 #####
3
4 ~~~ Node Name: Level_1 - Node Value: null
5   Attributes:
6     attr = 0
7
8 ##### LEVEL 2 #####
9
10 ~~~ Node Name: Level_2A - Node Value: null
11   Attributes:
12     attr = 1
13 ~~~ Node Name: Level_2B - Node Value: null
14   Attributes:
15     attr = 2
16 ~~~ Node Name: Level_2C - Node Value: null
17   Attributes:
18     attr = 3
19     attr2 = 4
20
21 ##### LEVEL 3 #####
22
23 ~~~ Node Name: Level_3AA - Node Value: text
24 ~~~ Node Name: Level_3AB - Node Value: text
25 ~~~ Node Name: Level_3AC - Node Value: text
26 ~~~ Node Name: Level_3AD - Node Value: text
27 ~~~ Node Name: Level_3BA - Node Value: text
28 ~~~ Node Name: Level_3BB - Node Value: text
29 ~~~ Node Name: Level_3BC - Node Value: text
30 ~~~ Node Name: Level_3BD - Node Value: text
31 ~~~ Node Name: Level_3CA - Node Value: text
32 ~~~ Node Name: Level_3CB - Node Value: text
33 ~~~ Node Name: Level_3CC - Node Value: text
34 ~~~ Node Name: Level_3CD - Node Value: text

```

Listing 6: Η έξοδος του προγράμματος

2

Στην συγκεκριμένη άσκηση ζητήθηκε η κατασκευή ενός κειμένου .xml το οποίο περιέχει το root , αντικείμενα με attributes και ένα επιπλέον παρακάτω επίπεδο χωρίς attributes. Το εν λόγω κείμενο .xml παρατίθεται παρακάτω:

```

1 <?xml version="1.0"?>
2 <?xml-stylesheet href="cars.xsl" type="text/xsl"?>
3 <vehicles>
4   <vehicle year="1995" make="Toyota" model="Prius">
5     <mileage>36500</mileage>
6     <color>Silver</color>
7     <price>$34100</price>
8   </vehicle>
9   <vehicle year="2000" make="Audi" model="TT Quattro">
10    <mileage>3550</mileage>
11    <color>Black</color>
12    <price>$41600</price>
13  </vehicle>
14  <vehicle year="2003" make="Citroen" model="C4 Picasso">
15    <mileage>25800</mileage>
16    <color>Cyan</color>
17    <price>$18900</price>
18  </vehicle>
19  <vehicle year="2005" make="BMW" model="F12">
20    <mileage>49300</mileage>
21    <color>White</color>
22    <price>$56995</price>
23  </vehicle>
24  <vehicle year="2008" make="Peugeot" model="308">
25    <mileage>14700</mileage>
26    <color>Blue</color>
27    <price>$22500</price>
28  </vehicle>
29 </vehicles>

```

Listing 7: Το κείμενο cars.xml

2.a .XSLT Transform

Στο συγκεκριμένο ερώτημα ζητήθηκε η κατασκευή μίας σελίδας html , η οποία θα αποτελείται από έναν πίνακα που θα έχει μία σειρά για κάθε αντικείμενο και μία στήλη για την τιμή του κάθε χαρακτηριστικού. Στην πρώτη σειρά φαίνεται ο τύπος του χαρακτηριστικού μαζί με ένα string "(a)" ή "(c)", ανάλογα αν το χαρακτηριστικό προέρχεται από attribute ή από παιδί του αντικειμένου αντίστοιχα.

Προκειμένου να επιτευχθεί ο μετασχηματισμός αυτός, χρειάστηκε η κατασκευή του ανάλογου κειμένου .xsl, το οποίο θα δώσει τις απαραίτητες οδηγίες στον .xslt transformer έτσι ώστε να κατασκευάσει την επιθυμητή σελίδα html. Το εν λόγω κείμενο .xsl παρατίθεται παρακάτω:

```

1 <?xml version="1.0"?>
2 <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3   <xsl:template match="/">
4     <html>
5       <head>
6         <title>Vehicles</title>
7       </head>
8
9       <body>
10        <h1 align="center">Vehicles</h1>

```



```

11      <table align="center" border="2px">
12          <tr>
13              <th>Year (a)</th>
14              <th>Make (a)</th>
15              <th>Model (a)</th>
16              <th>Mileage (c)</th>
17              <th>Color (c)</th>
18              <th>Price (c)</th>
19          </tr>
20          <xsl:for-each select="vehicles/vehicle">
21              <tr>
22                  <td><xsl:value-of select="@year"/></td>
23                  <td><xsl:value-of select="@make"/></td>
24                  <td><xsl:value-of select="@model"/></td>
25                  <td><xsl:value-of select="mileage"/></td>
26                  <td><xsl:value-of select="color"/></td>
27                  <td><xsl:value-of select="price"/></td>
28              </tr>
29          </xsl:for-each>
30      </table>
31  </body>
32 </html>
33 </xsl:template>
34 </xsl:stylesheet>

```

Listing 8: Το κείμενο cars.xsl

Εκτελώντας πλέον την παρακάτω εντολή:

```
java org.apache.xalan.xslt.Process -IN cars.xml -XSL cars.xsl -OUT cars.html
```

προκύπτει η ζητούμενη σελίδα html, της οποίας ο κώδικας παρουσιάζεται παρακάτω:

```

1 <html>
2 <head>
3 <META http-equiv="Content-Type" content="text/html; charset=UTF-8">
4 <title>Vehicles</title>
5 </head>
6 <body>
7 <h1 align="center">Vehicles</h1>
8 <table border="2px" align="center">
9 <tr>
10 <th>Year (a)</th><th>Make (a)</th><th>Model (a)</th><th>Mileage (c)</th><th>Color
    (c)</th><th>Price (c)</th>
11 </tr>
12 <tr>
13 <td>1995</td><td>Toyota</td><td>Prius</td><td>36500</td><td>Silver</td><td>$34100<
    /td>
14 </tr>
15 <tr>
16 <td>2000</td><td>Audi</td><td>TT Quattro</td><td>3550</td><td>Black</td><td>$41600
    </td>
17 </tr>
18 <tr>
19 <td>2003</td><td>Citroen</td><td>C4 Picasso</td><td>25800</td><td>Cyan</td><td>
    $18900</td>
20 </tr>
21 <tr>
22 <td>2005</td><td>BMW</td><td>F12</td><td>49300</td><td>White</td><td>$56995</td>
23 </tr>
24 <tr>

```

```

25 <td>2008</td><td>Peugeot</td><td>308</td><td>14700</td><td>Blue</td><td>$22500</td>
26 >
27 </tr>
28 </table>
29 </body>
</html>

```

Listing 9: Η σελίδα cars.html

Η απεικόνιση της παραπάνω σελίδας html σε έναν browser είναι η παρακάτω:

Vehicles

Year (a)	Make (a)	Model (a)	Mileage (c)	Color (c)	Price (c)
1995	Toyota	Prius	36500	Silver	\$34100
2000	Audi	TT Quattro	3550	Black	\$41600
2003	Citroen	C4 Picasso	25800	Cyan	\$18900
2005	BMW	F12	49300	White	\$56995
2008	Peugeot	308	14700	Blue	\$22500

Σχήμα 1: Η σελίδα cars.html, όπως απεικονίζεται σε έναν οποιονδήποτε browser.

2.b XML Transform Using Java Program

Στο συγκεκριμένο ερώτημα ζητήθηκε η ίδια λειτουργία όπως στο ερώτημα (2α), αλλά με χρήση προγράμματος Java. Για το λόγο αυτό χρησιμοποιήθηκε ο κώδικας του ερωτήματος (1β), έτσι ώστε να κατασκευαστεί η ζητούμενη σελίδα html. Ο κώδικας του εν λόγω προγράμματος παρατίθεται παρακάτω:

```

1 import org.xml.sax.*;
2 import org.xml.sax.helpers.*;
3 import javax.xml.parsers.*;
4 import java.util.ArrayList;
5 import java.io.PrintWriter;
6
7 public class XSLTransformer implements ContentHandler
8 {
9     public static class Node {
10         public String nodeName;
11         public String nodeValue;
12         public Attributes attr;
13         public Node(String name, Attributes attributes){
14             nodeName = name;
15             attr = attributes;
16         }
17     }
18     public static ArrayList<ArrayList<Node>> nodeList;
19     public static boolean foundTitle = false;
20     public static int a=-1;
21     private Locator lc;

```

```

22 public static void main(String[] args) throws Exception
23 {
24     XSLTransformer xmlep = new XSLTransformer();
25     nodeList = new ArrayList<ArrayList<Node>>();
26     PrintWriter output = new PrintWriter("cars.html");
27     xmlep.go(args[0]);
28     output.println("<html>");
29     output.println("<head>");
30     output.println("<META http-equiv=\"Content-Type\" content=\"text/html; charset=UTF-8\">");
31     output.println("<title>Vehicles</title>");
32     output.println("</head>");
33     output.println("<body>");
34     output.println("<h1 align=\"center\">Vehicles</h1>");
35     output.println("<table border=\"2px\" align=\"center\">");
36     output.println("<tr><th>Year (a)</th><th>Make (a)</th><th>Model (a)</th><th>Mileage (c)</th><th>Color (c)</th><th>Price (c)</th></tr>");
37     for (int i = 0 ; i < nodeList.get(1).size(); i++){
38         Node element = nodeList.get(1).get(i);
39         if (element.attr.getLength() > 0) {
40             output.println("<tr>");
41             for (int k = 0; k < element.attr.getLength(); k++) {
42                 output.println("<td>" + element.attr.getValue(k) + "</td>");
43             }
44         }
45         for (int j = i*3 ; j < i*3+3 && j < nodeList.get(2).size(); j++){
46             Node child_element = nodeList.get(2).get(j);
47             output.println("<td>" + child_element.nodeValue + "</td>");
48         }
49         output.println("</tr>");
50     }
51     output.println("</table>");
52     output.println("</body>");
53     output.println("</html>");
54     output.close();
55 }
56 public void go(String arg) throws Exception
57 {
58     SAXParserFactory spf = SAXParserFactory.newInstance();
59     SAXParser sp = spf.newSAXParser();
60     XMLReader sr = sp.getXMLReader();
61     sr.setContentHandler(this);
62     sr.parse(arg);
63 }
64 public void setDocumentLocator(Locator locator)
65 {
66     this.lc =locator ;
67 }
68 public void startDocument() throws SAXException {}
69 public void endDocument() throws SAXException {}
70 public void startPrefixMapping(String prefix, String uri) throws SAXException {}
71 public void endPrefixMapping(String prefix) throws SAXException {}
72 public void startElement(String namespaceURI, String localName, String qName,
73     Attributes atts) throws SAXException
74 {
75     a++;
76     if(nodeList.size()-1 < a) {
77         nodeList.add(new ArrayList<Node>());
78     }
79 }

```

```

77     }
78     nodeList.get(a).add(new Node(qName, new AttributesImpl(atts)));
79     foundTitle = true;
80 }
81 public void endElement(String namespaceURI, String localName, String qName)
82     throws SAXException
83 {
84     a--;
85 }
86 public void characters(char ch[], int start, int length) throws SAXException {
87     if (foundTitle){
88         nodeList.get(a).get(nodeList.get(a).size()-1).nodeValue = new String(
89 ch,start,length);
90         foundTitle = false;
91     }
92 }
93 public void ignorableWhitespace(char ch[], int start, int length) throws
94 SAXException {}
95 public void processingInstruction(String target, String data) throws
96 SAXException {}
97 public void skippedEntity(String name) throws SAXException {}
98 }

```

Listing 10: Το πρόγραμμα XSLTransformer.java

Εκτελώντας το παραπάνω πρόγραμμα με παράμετρο-είσοδο το κείμενο cars.xml, παράγεται ως έξοδος η ζητούμενη html σελίδα, όπως φαίνεται παρακάτω:

```

1 <html>
2 <head>
3 <META http-equiv="Content-Type" content="text/html; charset=UTF-8">
4 <title>Vehicles</title>
5 </head>
6 <body>
7 <h1 align="center">Vehicles</h1>
8 <table border="2px" align="center">
9 <tr><th>Year (a)</th><th>Make (a)</th><th>Model (a)</th><th>Mileage (c)</th><th>
    Color (c)</th><th>Price (c)</th></tr>
10 <tr>
11 <td>1995</td>
12 <td>Toyota</td>
13 <td>Prius</td>
14 <td>36500</td>
15 <td>Silver</td>
16 <td>$34100</td>
17 </tr>
18 <tr>
19 <td>2000</td>
20 <td>Audi</td>
21 <td>TT Quattro</td>
22 <td>3550</td>
23 <td>Black</td>
24 <td>$41600</td>
25 </tr>
26 <tr>
27 <td>2003</td>
28 <td>Citroen</td>
29 <td>C4 Picasso</td>
30 <td>25800</td>
31 <td>Cyan</td>
32 <td>$18900</td>

```

```

33 </tr>
34 <tr>
35 <td>2005</td>
36 <td>BMW</td>
37 <td>F12</td>
38 <td>49300</td>
39 <td>White</td>
40 <td>$56995</td>
41 </tr>
42 <tr>
43 <td>2008</td>
44 <td>Peugeot</td>
45 <td>308</td>
46 <td>14700</td>
47 <td>Blue</td>
48 <td>$22500</td>
49 </tr>
50 </table>
51 </body>
52 </html>

```

Listing 11: Η σελίδα cars.html

Η απεικόνιση της παραπάνω σελίδας html σε έναν browser είναι η παρακάτω:

Vehicles

Year (a)	Make (a)	Model (a)	Mileage (c)	Color (c)	Price (c)
1995	Toyota	Prius	36500	Silver	\$34100
2000	Audi	TT Quattro	3550	Black	\$41600
2003	Citroen	C4 Picasso	25800	Cyan	\$18900
2005	BMW	F12	49300	White	\$56995
2008	Peugeot	308	14700	Blue	\$22500

Σχήμα 2: Η σελίδα cars.html, όπως απεικονίζεται σε έναν οποιονδήποτε browser.