JEGYZŐKÖNYV

Adatkezelés XML környezetben

Online számítástechnikai áruház

Féléves feladat

**Készítette:** **Csomor Bence Patrik**

**Neptunkód:** **TVIK4I**

**Feladat leírása:**

A beadandóm témája egy online számítástechnikai áruház és a terméket vagy termékeket megvásárolni kívánó Vásárló által elindított folyamat bemutatása.

A Vásárló az Online Áruházban leadja a rendelését a kívánt Termék(ek)-re, amely az első lépése a folyamatnak. Ezt az áruház online weboldalán teheti meg a Vásárló.

Ezt követően az Online Áruház, mivel ez nem egy fizikai bolt, ezért minden esetben továbbítja a rendelést az adott Raktárnak.

A raktárakban tárolják az áruház termékeit cikkszámmal, árral, típusszámmal és kiadási évvel ellátva.

Amint megvan a megrendelt termék, a Vásárló vagy elmegy érte maga, vagy ha úgy dönt, házhoz is kiszállítják számára.

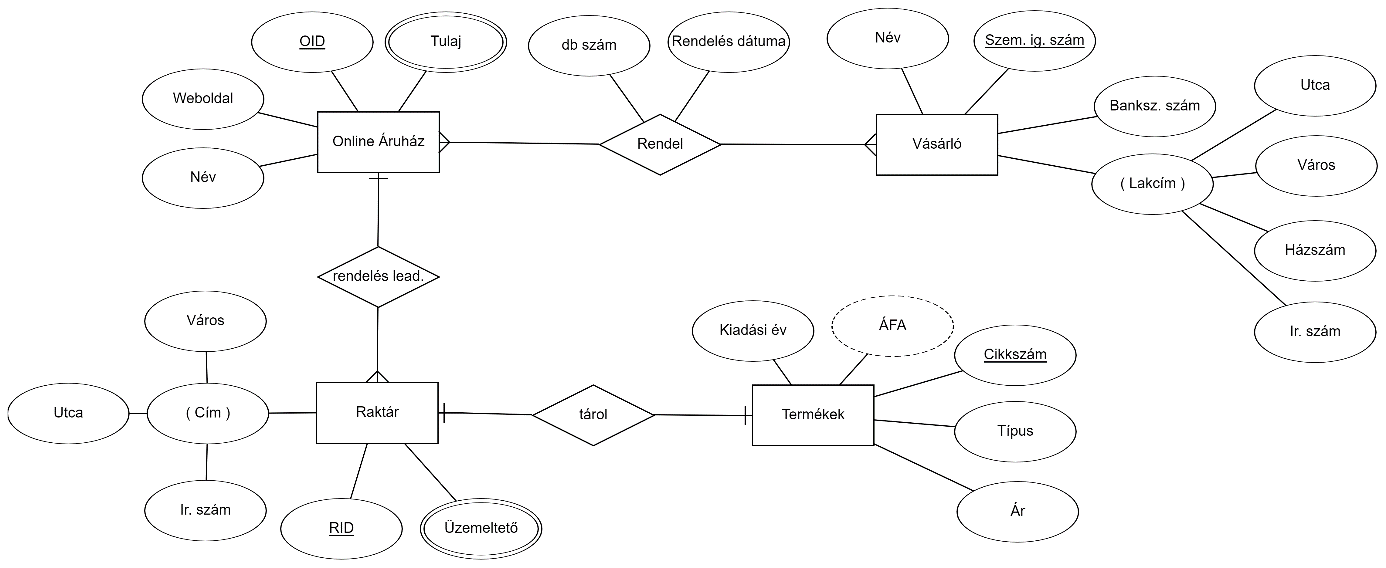
Kapcsolatok:

N:M 🡪 Több Vásárló lehet egy Online Áruházban és Több Online Áruházban vásárolhat egy Vásárló.

1:N 🡪 Egy raktár csak egy Online Áruházhoz tartozik, de több Raktára is lehet egy Online Áruháznak.

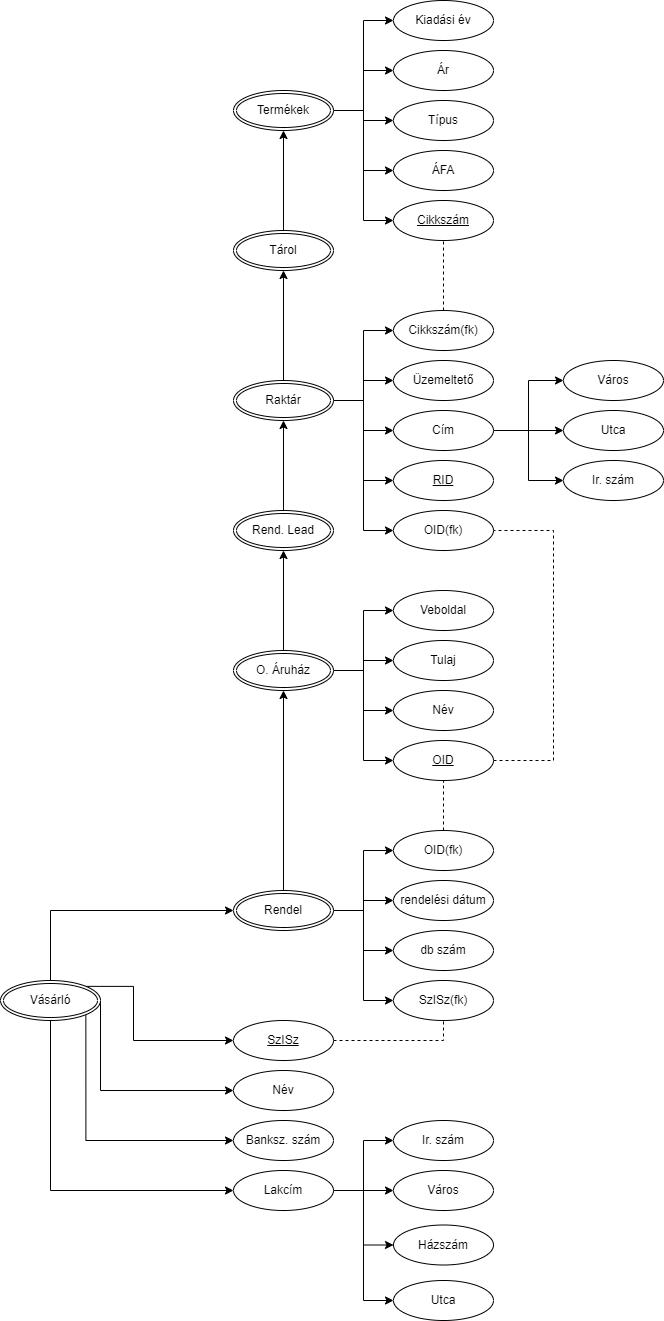
1:1 🡪 Egy terméket csak egy raktárban tudnak tárolni és egy raktárban egy adott terméket tárolnak el.

**ER modell:**



**XDM modell:**

Az alább látható XDM modell bemutatja fa struktúrában az egyedeket és attributumaikat a típusokkal együtt.



**XML dokumentum:**

<?xml version="1.0" encoding="UTF-8" ?>  
  
<rendeles>  
 <vasarlo SzISz="12345678">  
 <lakcim>  
 <utca>Pöttyös</utca>  
 <hazszam>69</hazszam>  
 <varos>Budapest</varos>  
 <iranyitoszam>1185</iranyitoszam>  
 </lakcim>  
 <bankszanlaSzam>123456789</bankszanlaSzam>  
 <nev>Akciós Áron</nev>  
 </vasarlo>  
 <vasarlo SzISz="87654321">  
 <lakcim>  
 <utca>Akác</utca>  
 <hazszam>420</hazszam>  
 <varos>Miskolc</varos>  
 <iranyitoszam>3654</iranyitoszam>  
 </lakcim>  
 <bankszanlaSzam>987654321</bankszanlaSzam>  
 <nev>Dil Emma</nev>  
 </vasarlo>  
 <vasarlo SzISz="12443121">  
 <lakcim>  
 <utca>Ametiszt</utca>  
 <hazszam>84</hazszam>  
 <varos>Budapest</varos>  
 <iranyitoszam>1186</iranyitoszam>  
 </lakcim>  
 <bankszanlaSzam>739274832</bankszanlaSzam>  
 <nev>Resz Elek</nev>  
 </vasarlo>  
 <onlineAruhaz OID="453id" SzISz="12345678">  
 <nev>Online Bolt</nev>  
 <weboldal>onlinebolt.hu</weboldal>  
 <tulaj>Online Ferenc</tulaj>  
 </onlineAruhaz>  
 <raktar RID="463id" SzISz="12345678">  
 <cim>  
 <varos>Miskolc</varos>  
 <utca>Eper</utca>  
 <iranyitoszam>3656</iranyitoszam>  
 </cim>  
 <uzemelteto>Üzemi Balázs</uzemelteto>  
 </raktar>  
 <onlineAruhaz OID="398id" SzISz="87654321">  
 <nev>Online Üzlet</nev>  
 <weboldal>onlineuzlet.hu</weboldal>  
 <tulaj>Online József</tulaj>  
 </onlineAruhaz>  
 <raktar RID="287id" SzISz="87654321">  
 <cim>  
 <varos>Budapest</varos>  
 <utca>Alma</utca>  
 <iranyitoszam>1421</iranyitoszam>  
 </cim>  
 <uzemelteto>Üzemi Tamás</uzemelteto>  
 </raktar>  
 <onlineAruhaz OID="837id" SzISz="12443121">  
 <nev>Online Kereskedés</nev>  
 <weboldal>onlinekereskedes.hu</weboldal>  
 <tulaj>Online József</tulaj>  
 </onlineAruhaz>  
 <raktar RID="280id" SzISz="12443121">  
 <cim>  
 <varos>Budapest</varos>  
 <utca>Körte</utca>  
 <iranyitoszam>1647</iranyitoszam>  
 </cim>  
 <uzemelteto>Üzemi Róbert</uzemelteto>  
 </raktar>  
 <termekek Cikkszam="298rt43" SzISz="12345678">  
 <afa>27</afa>  
 <tipus>valamifele34</tipus>  
 <ar>5000</ar>  
 <kiadasiEv>2015</kiadasiEv>  
 </termekek>  
 <termekek Cikkszam="436ju52" SzISz="87654321">  
 <afa>27</afa>  
 <tipus>masikfele87</tipus>  
 <ar>7000</ar>  
 <kiadasiEv>2014</kiadasiEv>  
 </termekek>  
 <termekek Cikkszam="938su28" SzISz="12443121">  
 <afa>27</afa>  
 <tipus>harmadikfele21</tipus>  
 <ar>15000</ar>  
 <kiadasiEv>2020</kiadasiEv>  
 </termekek>  
</rendeles>

**XML Schema:**

<?xml version="1.0" encoding="UTF-8"?>  
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="qualified">  
  
 <xs:element name="rendeles">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="vasarlo" type="vasarloType"/>  
 <xs:element name="onlineAruhaz" type="onlineAruhazType"/>  
 <xs:element name="raktar" type="raktarType"/>  
 <xs:element name="termekek" type="termekekType"/>  
 <xs:element name="keys">  
 <xs:key name="SzISz">  
 <xs:selector xpath="vasarloType" />  
 <xs:field xpath="SzISz" />  
 </xs:key>  
 <xs:keyref name="SzISzRef" refer="SzISz">  
 <xs:selector xpath="onlineAruhazType" />  
 <xs:field xpath="OID" />  
 </xs:keyref>  
 <xs:key name="OID">  
 <xs:selector xpath="onlineAruhazType" />  
 <xs:field xpath="OID" />  
 </xs:key>  
 <xs:keyref name="OIDRef" refer="OID">  
 <xs:selector xpath="vasarloType" />  
 <xs:field xpath="SzISz" />  
 </xs:keyref>  
 <xs:key name="RID">  
 <xs:selector xpath="raktarType" />  
 <xs:field xpath="RID" />  
 </xs:key>  
 <xs:keyref name="RIDRef" refer="RID">  
 <xs:selector xpath="onlineAruhazType" />  
 <xs:field xpath="OID" />  
 </xs:keyref>  
 <xs:key name="Cikkszam">  
 <xs:selector xpath="termekekType" />  
 <xs:field xpath="Cikkszam" />  
 </xs:key>  
 <xs:keyref name="CikkszamRef" refer="Cikkszam">  
 <xs:selector xpath="raktarType" />  
 <xs:field xpath="RID" />  
 </xs:keyref>  
 </xs:element>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 <xs:complexType name="vasarloType">  
 <xs:sequence>  
 <xs:element name="lakcim" type="lakcimType"/>  
 <xs:element name="bankszanlaSzam" type="xs:integer"/>  
 <xs:element name="nev" type="xs:string"/>  
 </xs:sequence>  
 <xs:attribute type="xs:string" name="id"/>  
 </xs:complexType>  
 <xs:complexType name="lakcimType">  
 <xs:sequence>  
 <xs:element name="utca" type="xs:string"/>  
 <xs:element name="hazszam" type="xs:integer"/>  
 <xs:element name="varos" type="xs:string"/>  
 <xs:element name="iranyitoszam" type="xs:integer"/>  
 </xs:sequence>  
 </xs:complexType>  
 <xs:complexType name="onlineAruhazType">  
 <xs:sequence>  
 <xs:element name="nev" type="xs:string"/>  
 <xs:element name="weboldal" type="xs:string"/>  
 <xs:element name="tulaj" type="xs:string"/>  
 </xs:sequence>  
 </xs:complexType>  
 <xs:complexType name="raktarType">  
 <xs:sequence>  
 <xs:element name="cim" type="cimType"/>  
 <xs:element name="uzemelteto" type="xs:string"/>  
 </xs:sequence>  
 </xs:complexType>  
 <xs:complexType name="cimType">  
 <xs:sequence>  
 <xs:element name="varos" type="xs:string"/>  
 <xs:element name="utca" type="xs:string"/>  
 <xs:element name="iranyitoszam" type="xs:integer"/>  
 </xs:sequence>  
 </xs:complexType>  
 <xs:complexType name="termekekType">  
 <xs:sequence>  
 <xs:element name="afa" type="xs:integer"/>  
 <xs:element name="tipus" type="xs:string"/>  
 <xs:element name="ar" type="xs:integer"/>  
 <xs:element name="kiadasiEv" type="xs:integer"/>  
 </xs:sequence>  
 </xs:complexType>  
</xs:schema>

**DOM Adatolvasás:**

Itt kiíratom a fájl adatait az XML Schema alapján fa struktúrában a konzolra.

package hu.domparse.tvik4i;  
  
import org.w3c.dom.\*;  
import org.xml.sax.\*;  
  
import javax.xml.parsers.DocumentBuilder;  
import javax.xml.parsers.DocumentBuilderFactory;  
import javax.xml.parsers.ParserConfigurationException;  
import java.io.File;  
import java.io.IOException;  
  
public class DOMReadTVIK4I {  
 public static void main(String[] args) throws SAXException, IOException, ParserConfigurationException {  
 //Létrehozás  
 File xmlFile = new File("src/hu/domparse/tvik4i/XMLTVIK4I.xml");  
 //Building és konvertálás  
 DocumentBuilderFactory factory = DocumentBuilderFactory.*newInstance*();  
 DocumentBuilder documentBuilder = factory.newDocumentBuilder();  
 Document document = documentBuilder.parse(xmlFile);  
 //Normalizálás  
 document.getDocumentElement().normalize();  
 //Gyökérelement kiírása  
 System.*out*.println(("Gyökérelement: " + document.getDocumentElement().getNodeName()));  
 //Vásárló elementek nodeList-be illesztése  
 NodeList nodeList = document.getElementsByTagName("vasarlo");  
 System.*out*.println("----------------------------------------------");  
 //Kiírás ciklusa, nodeList/nNode-on végighaladva az  
 // elkészített algoritmusokkal kiírja a kért adatokat  
 for (int i = 0; i < nodeList.getLength(); i++) {  
 Node nNode = nodeList.item(i);  
 if (nNode.getNodeType() == Node.*ELEMENT\_NODE*) {  
 Element element = (Element) nNode;  
 String SzISz = element.getAttribute("SzISz");  
 String OID = element.getAttribute("OID");  
 //Adatok átadása a node-okba  
 Node node1 = element.getElementsByTagName("nev").item(0);  
 String nev = node1.getTextContent();  
 Node node2 = element.getElementsByTagName("lakcim").item(0);  
 String lakcim = node2.getTextContent();  
 Node node3 = element.getElementsByTagName("bankszanlaSzam").item(0);  
 String banksz = node3.getTextContent();  
 //Kiírás  
 System.*out*.println("Vásárló neve: " + nev);  
 System.*out*.println("Vásárló szem. ig. száma: " + SzISz);  
 System.*out*.println("Lakcíme: " + lakcim);  
 System.*out*.println("Bankszámla száma: " + banksz + "\n");  
 //Metódus hívások  
 *listOnlineAruhaz*(document, OID);  
 System.*out*.println("Raktár:\n" );  
 *listRaktar*(document, SzISz);  
 System.*out*.println(nev + " Terméke:\n" );  
 *listTermekek*(document, SzISz);  
 System.*out*.println("\n");  
 }  
 }  
 }  
 //Online Áruház adatainak kiírása az elõzõ módszer alapján  
 public static void listOnlineAruhaz (Document document, String OID) {  
 NodeList nodeList = document.getElementsByTagName("onlineAruhaz");  
 for (int i = 0; i < nodeList.getLength(); i++) {  
 Node nNode = nodeList.item(i);  
 if (nNode.getNodeType() == Node.*ELEMENT\_NODE*) {  
 Element element = (Element) nNode;  
 if (element.getAttribute("OID").equals(OID)) {  
 Node node1 = element.getElementsByTagName("nev").item(0);  
 String nev = node1.getTextContent();  
 Node node2 = element.getElementsByTagName("weboldal").item(0);  
 String weboldal = node2.getTextContent();  
 Node node3 = element.getElementsByTagName("tulaj").item(0);  
 String tulaj = node3.getTextContent();  
 System.*out*.println("Online áruház neve: " + nev);  
 System.*out*.println("Online áruház weboldala: " + weboldal);  
 System.*out*.println("Online áruház tulaja: " + tulaj + "\n");  
 }  
 }  
 }  
 }  
 //Raktár adatainak kiírása az elõzõ módszer alapján  
 public static void listRaktar (Document document, String SzISz) {  
 NodeList nodeList = document.getElementsByTagName("raktar");  
 for (int i = 0; i < nodeList.getLength(); i++) {  
 Node nNode = nodeList.item(i);  
 if (nNode.getNodeType() == Node.*ELEMENT\_NODE*) {  
 Element element = (Element) nNode;  
 if (element.getAttribute("SzISz").equals(SzISz)) {  
 String rid = element.getAttribute("RID");  
 Node node1 = element.getElementsByTagName("uzemelteto").item(0);  
 String uzemelteto = node1.getTextContent();  
 Node node2 = element.getElementsByTagName("cim").item(0);  
 String cim = node2.getTextContent();  
 System.*out*.println("Raktár id-je: " + rid);  
 System.*out*.println("Üzemeltető neve: " + uzemelteto);  
 System.*out*.println("Raktár címe: " + cim + "\n");  
 }  
 }  
 }  
 }  
 //Termék(ek) adatainak kiírása az elõzõ módszer alapján  
 public static void listTermekek (Document document, String SzISz) {  
 NodeList nodeList = document.getElementsByTagName("termekek");  
 for (int i = 0; i < nodeList.getLength(); i++) {  
 Node nNode = nodeList.item(i);  
 if (nNode.getNodeType() == Node.*ELEMENT\_NODE*) {  
 Element element = (Element) nNode;  
 if (element.getAttribute("SzISz").equals(SzISz)) {  
 String cikkszam = element.getAttribute("Cikkszam");  
 Node node1 = element.getElementsByTagName("afa").item(0);  
 String afa = node1.getTextContent();  
 Node node2 = element.getElementsByTagName("tipus").item(0);  
 String tipus = node2.getTextContent();  
 Node node3 = element.getElementsByTagName("ar").item(0);  
 String ar = node3.getTextContent();  
 Node node4 = element.getElementsByTagName("kiadasiEv").item(0);  
 String kiadasiEv = node4.getTextContent();  
 System.*out*.println("Termék cikkszáma: " + cikkszam);  
 System.*out*.println("Termék áfája: " + afa);  
 System.*out*.println("Termék tipusa: " + tipus);  
 System.*out*.println("Termék ára: " + ar);  
 System.*out*.println("Termék kiadási éve: " + kiadasiEv + "\n");  
 }  
 }  
 }  
 }  
}

**DOM Adatmódosítás:**

Itt módosítom a vevő nevét, a termék típusát és árát is.

package hu.domparse.tvik4i;  
  
import org.w3c.dom.\*;  
import org.xml.sax.\*;  
  
import javax.xml.parsers.DocumentBuilder;  
import javax.xml.parsers.DocumentBuilderFactory;  
import javax.xml.parsers.ParserConfigurationException;  
import javax.xml.transform.Transformer;  
import javax.xml.transform.TransformerConfigurationException;  
import javax.xml.transform.TransformerException;  
import javax.xml.transform.TransformerFactory;  
import javax.xml.transform.dom.DOMSource;  
import javax.xml.transform.stream.StreamResult;  
import java.io.File;  
import java.io.IOException;  
import java.util.Scanner;  
  
public class DOMModifyTVIK4I {  
 public static void main(String[] args) throws SAXException, IOException, ParserConfigurationException, TransformerConfigurationException, TransformerException {  
 //Az eredeti XML fájl és a megváltoztatott xml fájl megadása  
 File xmlFile = new File("src/hu/domparse/tvik4i/XMLTVIK4I.xml");  
 File xmlFileMODIFIED = new File("src/hu/domparse/tvik4i/XMLTVIK4IMODIFIED.xml");  
 //Scanner nyitása a beolvasáshoz  
 Scanner sc = new Scanner(System.*in*);  
 //Builder + konvertálás  
 DocumentBuilderFactory factory = DocumentBuilderFactory.*newInstance*();  
 DocumentBuilder documentBuilder = factory.newDocumentBuilder();  
 Document document = documentBuilder.parse(xmlFile);  
 //Normalizálás  
 document.getDocumentElement().normalize();  
 //A vásárló elemek kiválasztása.  
 NodeList nodeList = document.getElementsByTagName("vasarlo");  
 NodeList nodeList2 = document.getElementsByTagName("termekek");  
// NodeList nodeList3 = document.getElementsByTagName("vasarlo");  
 //ciklus a változtatásokhoz  
 for (int i = 0; i < nodeList.getLength(); i++) {  
 Node nNode = nodeList.item(i);  
 Node nNode2 = nodeList2.item(i);  
 Element element = (Element) nNode;  
 Element element2 = (Element) nNode2;  
 //Név bekérése  
 Node node1 = element.getElementsByTagName("nev").item(0);  
 String nev = node1.getTextContent();  
 Node node2 = element2.getElementsByTagName("tipus").item(0);  
 String tipus = node2.getTextContent();  
 Node node3 = element2.getElementsByTagName("ar").item(0);  
 String ar = node3.getTextContent();  
 //Átadás a node-nak  
 System.*out*.println("A vásárló jelenlegi neve:" + nev);  
 System.*out*.println("Add meg a vásárló új nevét: \n");  
 //Bekérés billentyûzeten  
 String modifiedname = sc.nextLine();  
 //Beállítás node-on keresztül  
 node1.setTextContent(modifiedname);  
  
 //Termék típusának és árának módosítása  
 System.*out*.println("Termék jelenlegi típusa:" + tipus);  
 System.*out*.println("Add meg a termék új típusát:");  
 String newTipus = sc.nextLine();  
 node2.setTextContent(newTipus);  
 System.*out*.println("Termék jelenlegi ára:" + ar);  
 System.*out*.println("Add meg a termék új árát:");  
 String newAr = sc.nextLine();  
 node3.setTextContent(newAr);  
  
 }  
  
 //Scanner zárása  
 sc.close();  
 //transformer és domsource használatával változtatjuk a fájlt  
 TransformerFactory transformerFactory = TransformerFactory.*newInstance*();  
 Transformer transformer = transformerFactory.newTransformer();  
 DOMSource source = new DOMSource(document);  
 //A változtatás a result-ba került  
 StreamResult result = new StreamResult(xmlFileMODIFIED);  
 //Beírásra kerül a módosított fájlba a módosítás  
 transformer.transform(source, result);  
 }  
}

**DOM adatlekérdezés:**

Itt lekérdezem a vásárlók neveit, az online áruházak neveit és webcímüket is.

package hu.domparse.tvik4i;  
  
import org.w3c.dom.Document;  
import org.w3c.dom.Element;  
import org.w3c.dom.Node;  
import org.w3c.dom.NodeList;  
import org.xml.sax.SAXException;  
  
import javax.xml.parsers.DocumentBuilder;  
import javax.xml.parsers.DocumentBuilderFactory;  
import javax.xml.parsers.ParserConfigurationException;  
import java.io.File;  
import java.io.IOException;  
  
public class DOMQueryTVIK4I {  
 public static void main(String[] args) throws SAXException, IOException, ParserConfigurationException {  
 try{  
 //Létrehozás  
 File xmlFile = new File("src/hu/domparse/tvik4i/XMLTVIK4I.xml");  
 //Building és konvertálás  
 DocumentBuilderFactory factory = DocumentBuilderFactory.*newInstance*();  
 DocumentBuilder documentBuilder = factory.newDocumentBuilder();  
 Document document = documentBuilder.parse(xmlFile);  
 //Normalizálás  
 document.getDocumentElement().normalize();  
 //Gyökérelement kiírása  
 System.*out*.println(("Gyökérelement: " + document.getDocumentElement().getNodeName()));  
 //Vásárló elementek nodeList-be illesztése  
 NodeList nodeList = document.getElementsByTagName("vasarlo");  
 NodeList nodeList2 = document.getElementsByTagName("onlineAruhaz");  
 System.*out*.println("----------------------------------------------");  
 for (int temp = 0; temp < nodeList.getLength(); temp++) {  
 Node nNode = nodeList.item(temp);  
  
 if (nNode.getNodeType() == Node.*ELEMENT\_NODE*) {  
 Element element = (Element) nNode;  
 System.*out*.print("\n" + (temp+1) + ". Vásárló:");  
 System.*out*.println(element.getAttribute("vasarlo"));  
 NodeList vasarloNameList = element.getElementsByTagName("nev");  
  
 for (int count = 0; count < vasarloNameList.getLength(); count++) {  
 Node node1 = vasarloNameList.item(count);  
  
 if (node1.getNodeType() == node1.*ELEMENT\_NODE*) {  
 Element vasarlo = (Element) node1;  
 System.*out*.print("neve : ");  
 System.*out*.println(vasarlo.getTextContent());  
 }  
 }  
 }  
 }  
 for (int temp = 0; temp < nodeList2.getLength(); temp++) {  
 Node nNode = nodeList2.item(temp);  
  
 if (nNode.getNodeType() == Node.*ELEMENT\_NODE*) {  
 Element element = (Element) nNode;  
 System.*out*.print("\n" + (temp+1) + ". Online Áruház:");  
 System.*out*.println(element.getAttribute("onlineAruhaz"));  
 NodeList oAruhazNameList = element.getElementsByTagName("nev");  
 NodeList oAruhazWebList = element.getElementsByTagName("weboldal");  
  
 for (int count = 0; count < oAruhazNameList.getLength(); count++) {  
 Node node1 = oAruhazNameList.item(count);  
 Node node2 = oAruhazWebList.item(count);  
  
 if (node1.getNodeType() == node1.*ELEMENT\_NODE*) {  
 Element onlineAruhaz = (Element) node1;  
 System.*out*.print("neve : ");  
 System.*out*.println(onlineAruhaz.getTextContent());  
 }  
 if (node2.getNodeType() == node2.*ELEMENT\_NODE*) {  
 Element onlineAruhaz = (Element) node2;  
 System.*out*.print("webcíme : ");  
 System.*out*.println(onlineAruhaz.getTextContent());  
 }  
 }  
 }  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
}