Jegyzőkönyv

Adatkezelés XML környezetben

Féléves Feladat

Készítette: **Baranyi Gábor**

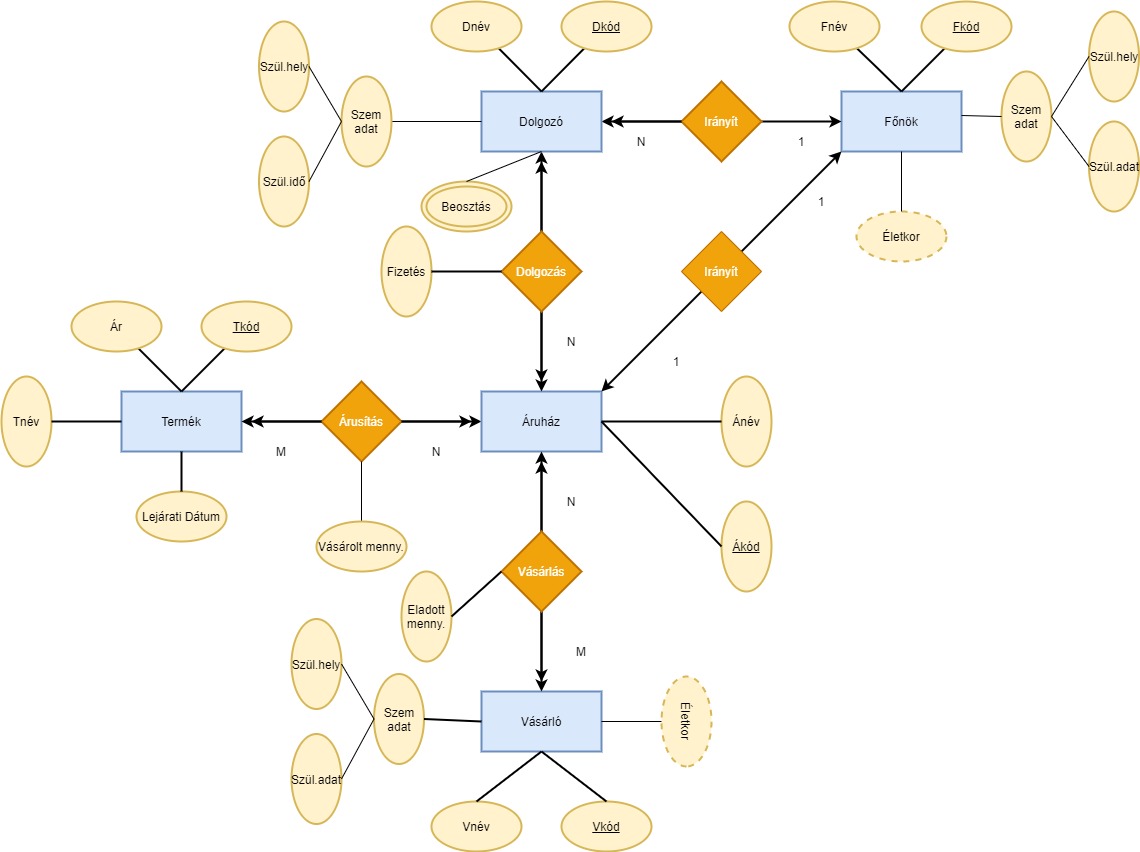
Neptunkód: **CRC7FC**

**A feladat leírása:**

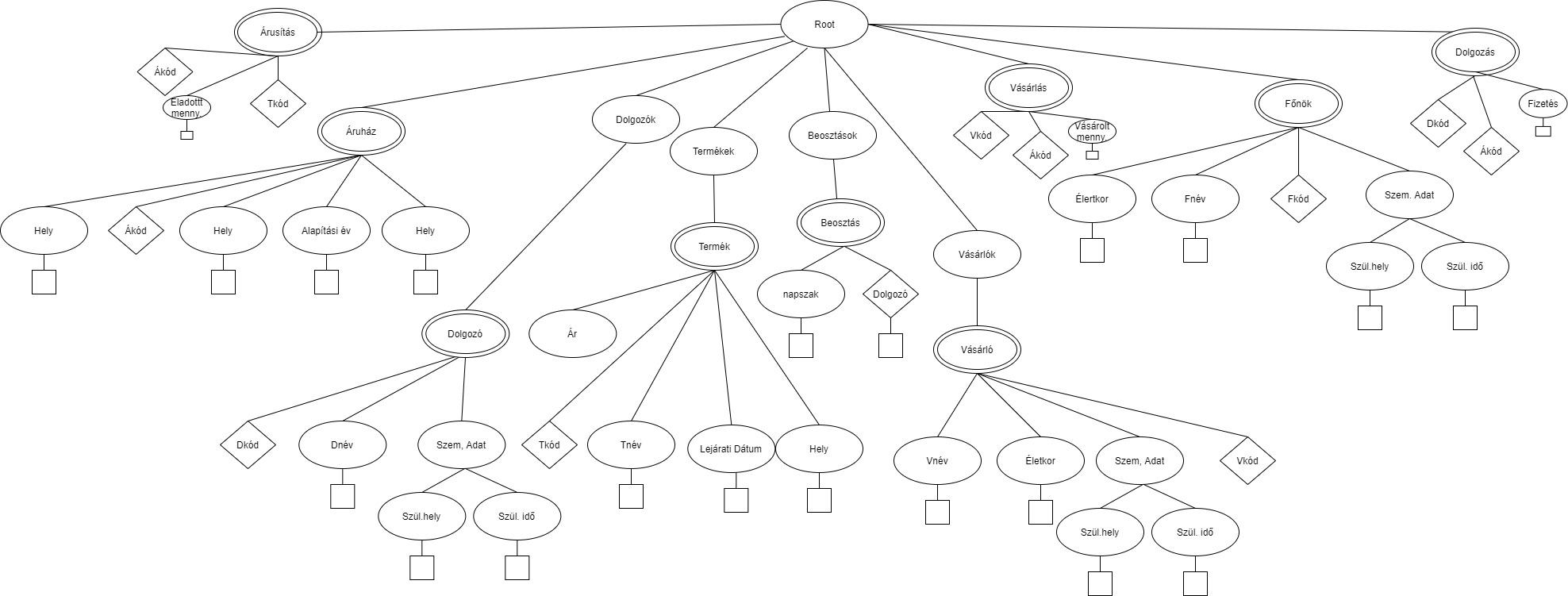
A beadandó tematikája egy áruház és annak egységei, melyek a dolgozók, a főnök, a termék és a vásárló. Az Áruháznak ugyebár magától értetődően vannak dolgozói, akik ott dolgoznak, vásárlói, akik ott termékeket vásárolnak, illetve termék, amiket árusít a vásárlóknak az áruház.

Az Áruháznak és a dolgozónak van egy főnöke, melynek szerepe, hogy irányítsa a dolgozók munkáját és az áruházat. A vásárlóknak és a főnöknek van egy kiszámítható tulajdonsága, ami nem más, mint az életkor. Ezt úgy számolhatjuk ki, ha kivonjuk a jelenlegi dátumból a vásárló vagy a főnök születési dátumát.

A dolgozónak, a főnöknek és a vásárlónak egyaránt a Szem.Adat egy összetett tulajdonság, amely magába foglalja a Szül.helyet és az Szül.időt is. A terméknek van egy vonalkódja ami az egyedi kódja (Tkód), van neve és ára illetve lejárti ideje. A főnöknek, a vásárlóknak és Dolgozónak szintén van neve és egyedi azonosítókódja. Ezen két tulajdonsága magától értetődően meg van az áruháznak is. Az áruház továbbá tartalmaz egy alapítási évet és helyet is.

**ER-model: **

**XDM modell:**

E-R modellt XDM modellre úgy konvertálhatunk, hogy három jelölést kell használni. Van az elem, amely az XDM modellben ellipszissel van rajzolva. Van az attribútum, jelölése rombusz, szöveges tartalmat téglalappal jelöljük. Ezeken kívül, ha egy elem többszörös előfordulási lehetőséggel bír azt dupla ellipszissel jelöljük. A több a többhöz kapcsolatokat külön meg kell mutatni.

**Xml dokumentum:**

<?xml version="1.0" encoding="UTF-8"?>  
<root>  
 <Főnök fkód="11">  
 <Fnév>Baranyi Gábor</Fnév>  
 <Életkor>20</Életkor>  
 <Szemadat>  
 <Szülhely>Miskolc</Szülhely>  
 <Szülidő>2001.03.22</Szülidő>  
 </Szemadat>  
 </Főnök>  
 <Áruház Ákód="21">  
 <Alapításiév>1989</Alapításiév>  
 <Hely>Miskolc</Hely>  
 <Ánév>Tesco</Ánév>  
 <Dolgozók fizetés="1700ft/h">  
 <Dolgozó Dkód="31" Ákód="21">  
 <Dnév>Nagy János</Dnév>  
 <Szemadat>  
 <Szülhely>Miskolc</Szülhely>  
 <Szülidő>2000.12.02</Szülidő>  
 </Szemadat>  
 </Dolgozó>  
 <Dolgozó Dkód="32" Ákód="21">  
 <Dnév>Kis Beáta</Dnév>  
 <Szemadat>  
 <Szülhely>Eger</Szülhely>  
 <Szülidő>2002.02.30</Szülidő>  
 </Szemadat>  
 </Dolgozó>  
 <Dolgozó Dkód="33" Ákód="21">  
 <Dnév>Mákos Maja</Dnév>  
 <Szemadat>  
 <Szülhely>Miskolc</Szülhely>  
 <Szülidő>1999.06.17</Szülidő>  
 </Szemadat>  
 </Dolgozó>  
 <Dolgozó Dkód="34" Ákód="21">  
 <Dnév>Kiss Márton</Dnév>  
 <Szemadat>  
 <Szülhely>Miskolc</Szülhely>  
 <Szülidő>1990.12.02</Szülidő>  
 </Szemadat>  
 </Dolgozó>  
 <Dolgozó Dkód="35" Ákód="21">  
 <Dnév>Nagy Áron</Dnév>  
 <Szemadat>  
 <Szülhely>Eger</Szülhely>  
 <Szülidő>1997.02.30</Szülidő>  
 </Szemadat>  
 </Dolgozó>  
 <Dolgozó Dkód="36" Ákód="21">  
 <Dnév>Dió Brando</Dnév>  
 <Szemadat>  
 <Szülhely>Miskolc</Szülhely>  
 <Szülidő>1967.06.17</Szülidő>  
 </Szemadat>  
 </Dolgozó>  
 <Dolgozó Dkód="37" Ákód="21">  
 <Dnév>Kujo Jotaro</Dnév>  
 <Szemadat>  
 <Szülhely>Miskolc</Szülhely>  
 <Szülidő>1999.05.12</Szülidő>  
 </Szemadat>  
 </Dolgozó>  
 </Dolgozók>  
 <Beosztások>  
 <Beosztás Dolgozó="31">  
 <napszak>Nappal</napszak>  
 </Beosztás>  
 <Beosztás Dolgozó="32">  
 <napszak>Nappal</napszak>  
 </Beosztás>  
 <Beosztás Dolgozó="33">  
 <napszak>Éjszaka</napszak>  
 </Beosztás>  
 <Beosztás Dolgozó="34">  
 <napszak>Éjszaka</napszak>  
 </Beosztás>  
 <Beosztás Dolgozó="35">  
 <napszak>Nappal</napszak>  
 </Beosztás>  
 <Beosztás Dolgozó="36">  
 <napszak>Éjszaka</napszak>  
 </Beosztás>  
 <Beosztás Dolgozó="37">  
 <napszak>Nappal</napszak>  
 </Beosztás>  
 </Beosztások>  
 <Vásárlók Vásároltmennyiség="184db">  
 <Vásárló Vkód="41" Ákód="21">  
 <Vnév>Igazságos Magdolna</Vnév>  
 <Életkor>35</Életkor>  
 <Szemadat>  
 <Szülhely>Eger</Szülhely>  
 <Szülidő>1986.06.02</Szülidő>  
 </Szemadat>  
 </Vásárló>  
 <Vásárló Vkód="42" Ákód="21">  
 <Vnév>Hazug Károly</Vnév>  
 <Életkor>60</Életkor>  
 <Szemadat>  
 <Szülhely>Szolnok</Szülhely>  
 <Szülidő>1961.07.10</Szülidő>  
 </Szemadat>  
 </Vásárló>  
 <Vásárló Vkód="43" Ákód="21">  
 <Vnév>Mikorka Kálmán</Vnév>  
 <Életkor>15</Életkor>  
 <Szemadat>  
 <Szülhely>Miskolc</Szülhely>  
 <Szülidő>2006.09.18</Szülidő>  
 </Szemadat>  
 </Vásárló>  
 <Vásárló Vkód="44" Ákód="21">  
 <Vnév>Jakab Kitti</Vnév>  
 <Életkor>32</Életkor>  
 <Szemadat>  
 <Szülhely>Eger</Szülhely>  
 <Szülidő>1989.06.02</Szülidő>  
 </Szemadat>  
 </Vásárló>  
 <Vásárló Vkód="45" Ákód="21">  
 <Vnév>Atya Zsozé</Vnév>  
 <Életkor>37</Életkor>  
 <Szemadat>  
 <Szülhely>Eger</Szülhely>  
 <Szülidő>1984.06.02</Szülidő>  
 </Szemadat>  
 </Vásárló>  
 <Vásárló Vkód="46" Ákód="21">  
 <Vnév>Nagy Lali</Vnév>  
 <Életkor>30</Életkor>  
 <Szemadat>  
 <Szülhely>Szolnok</Szülhely>  
 <Szülidő>1991.07.10</Szülidő>  
 </Szemadat>  
 </Vásárló>  
 <Vásárló Vkód="47" Ákód="21">  
 <Vnév>Mixáth Julio</Vnév>  
 <Életkor>17</Életkor>  
 <Szemadat>  
 <Szülhely>Miskolc</Szülhely>  
 <Szülidő>2004.09.18</Szülidő>  
 </Szemadat>  
 </Vásárló>  
 <Vásárló Vkód="48" Ákód="21">  
 <Vnév>Kaneki Ken</Vnév>  
 <Életkor>19</Életkor>  
 <Szemadat>  
 <Szülhely>Eger</Szülhely>  
 <Szülidő>2002.06.02</Szülidő>  
 </Szemadat>  
 </Vásárló>  
 </Vásárlók>  
 <Termékek Eladottmennyiség="185db">  
 <Termék Tkód="51" Ákód="21">  
 <Tnév>Falnijó virsli</Tnév>  
 <LejáratiDátum>2021.12.04</LejáratiDátum>  
 <ár>280Ft</ár>  
 </Termék>  
 <Termék Tkód="52" Ákód="21">  
 <Tnév>Lays chips</Tnév>  
 <LejáratiDátum>2022.01.14</LejáratiDátum>  
 <ár>300Ft</ár>  
 </Termék>  
 <Termék Tkód="53" Ákód="21">  
 <Tnév>Kenyér</Tnév>  
 <LejáratiDátum>2021.11.28</LejáratiDátum>  
 <ár>190Ft</ár>  
 </Termék>  
 <Termék Tkód="54" Ákód="21">  
 <Tnév>Head and shoulders Sampon</Tnév>  
 <LejáratiDátum>2022.10.12</LejáratiDátum>  
 <ár>980Ft</ár>  
 </Termék>  
 <Termék Tkód="55" Ákód="21">  
 <Tnév>BB pezsgő</Tnév>  
 <LejáratiDátum>2022.12.04</LejáratiDátum>  
 <ár>1980Ft</ár>  
 </Termék>  
 <Termék Tkód="56" Ákód="21">  
 <Tnév>Cheetos</Tnév>  
 <LejáratiDátum>2021.12.04</LejáratiDátum>  
 <ár>280Ft</ár>  
 </Termék>  
 <Termék Tkód="57" Ákód="21">  
 <Tnév>Párizsi rúd</Tnév>  
 <LejáratiDátum>2022.01.14</LejáratiDátum>  
 <ár>600Ft</ár>  
 </Termék>  
 <Termék Tkód="58" Ákód="21">  
 <Tnév>Májkrém</Tnév>  
 <LejáratiDátum>2021.11.28</LejáratiDátum>  
 <ár>340Ft</ár>  
 </Termék>  
 <Termék Tkód="58" Ákód="21">  
 <Tnév>Sonka rúd</Tnév>  
 <LejáratiDátum>2022.10.12</LejáratiDátum>  
 <ár>980Ft</ár>  
 </Termék>  
 </Termékek>  
 </Áruház>  
</root>

**XSD schema XML dokumentumból:**

<?xml version="1.0" encoding="UTF-8"?>  
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  
 <xs:element name="root">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="Főnök">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="Fnév" type="xs:string"/>  
 <xs:element name="Életkor" type ="xs:string"/>  
 <xs:element name="Szemadat">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="Szülhely" type="xs:string"/>  
 <xs:element name="Szülidő" type="ido"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 <xs:attribute name="fkód" type="xs:string"/>  
 </xs:complexType>  
 </xs:element>  
 <xs:element name="Áruház">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="Alapításiév" type="xs:positiveInteger"/>  
 <xs:element name="Hely" type="xs:string"/>  
 <xs:element name="Ánév" type="xs:string"/>  
 <xs:element name="Dolgozók">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="Dolgozó" maxOccurs="unbounded">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="Dnév" type="xs:attribute"/>  
 <xs:element name="Szemadat">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="Szülhely" type="xs:string"/>  
 <xs:element name="Szülidő" type="ido"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 <xs:attribute name="Ákód" type="xs:string"/>  
 <xs:attribute name="Dkód" type="xs:string"/>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 <xs:attribute name="fizetés" type="xs:string"/>  
 </xs:complexType>  
 </xs:element>  
 <xs:element name="Beosztások">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="Beosztás" maxOccurs="unbounded">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="napszak" type="xs:string"/>  
 </xs:sequence>  
 <xs:attribute name="Dolgozó" type="xs:string"/>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 <xs:element name="Vásárlók">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="Vásárló" maxOccurs="unbounded">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="Vnév" type="xs:string"/>  
 <xs:element name="Életkor" type="xs:positiveInteger"/>  
 <xs:element name="Szemadat">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="Szülhely" type="xs:string"/>  
 <xs:element name="Szülidő" type="ido"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 <xs:attribute name="Vkód" type="xs:string"/>  
 <xs:attribute name="Ákód" type="xs:string"/>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 <xs:attribute name="Vásároltmennyiség" type="xs:string"/>  
 </xs:complexType>  
 </xs:element>  
 <xs:element name="Termékek">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="Termék" maxOccurs="unbounded">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="Tnév" type="xs:string"/>  
 <xs:element name="LejáratiDátum" type="ido"/>  
 <xs:element name="ár" type="xs:string"/>  
 </xs:sequence>  
 <xs:attribute name="Tkód" type="xs:string"/>  
 <xs:attribute name="Ákód" type="xs:string"/>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 <xs:attribute name="Eladottmennyiség" type="xs:string"/>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 <xs:attribute name="Ákód" type="xs:string"/>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 </xs:complexType>  
 <xs:key name="Á\_key">  
 <xs:selector xpath="Áruház"/>  
 <xs:field xpath="@Ákód"/>  
 </xs:key>  
 <xs:key name="D\_key">  
 <xs:selector xpath="Dolgozó"/>  
 <xs:field xpath="@Dkód"/>  
 </xs:key>  
 <xs:key name="T\_key">  
 <xs:selector xpath="Termék"/>  
 <xs:field xpath="@Dkód"/>  
 </xs:key>  
 <xs:key name="V\_key">  
 <xs:selector xpath="Vásárló"/>  
 <xs:field xpath="@Vkód"/>  
 </xs:key>  
 <xs:keyref name="Termék\_ákód" refer="Á\_key">  
 <xs:selector xpath="Termék"/>  
 <xs:field xpath="@Ákód"/>  
 </xs:keyref>  
 <xs:keyref name="Dolgozó\_ákód" refer="Á\_key">  
 <xs:selector xpath="Dolgozó"/>  
 <xs:field xpath="@Ákód"/>  
 </xs:keyref>  
 <xs:keyref name="Vásárló\_ákód" refer="Á\_key">  
 <xs:selector xpath="Vásárló"/>  
 <xs:field xpath="@Ákód"/>  
 </xs:keyref>  
 </xs:element>  
 <xs:simpleType name="ido">  
 <xs:restriction base="xs:string">  
 <xs:pattern value="\d{4}[.]\d{2}[.]\d2}"/>  
 <xs:length value="10"/>  
 </xs:restriction>  
 </xs:simpleType>  
</xs:schema>

**DOM modifier:**

Átírja a Főnök nevét és kiírja egy dokumentumba.

package hu.domparse.crc7fc;  
  
import java.io.File;  
  
import javax.xml.parsers.DocumentBuilder;  
import javax.xml.parsers.DocumentBuilderFactory;  
import javax.xml.parsers.ParserConfigurationException;  
import javax.xml.transform.Result;  
import javax.xml.transform.Source;  
import javax.xml.transform.Transformer;  
import javax.xml.transform.TransformerFactory;  
import javax.xml.transform.dom.DOMSource;  
import javax.xml.transform.stream.StreamResult;  
  
import org.w3c.dom.Document;  
import org.w3c.dom.Element;  
import org.w3c.dom.Node;  
import org.w3c.dom.NodeList;  
  
public class DomModifycrc7fc {  
 public static void main(String[] args) {  
 NodeList list;  
 try {  
 DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();  
 DocumentBuilder builder = factory.newDocumentBuilder();  
 // fájl beolvasása  
 Document document = builder.parse(new File("XMLcrc7fc.xml"));  
 document.getDocumentElement().normalize();  
 // Aktuális elem meghatározása  
 list = document.getElementsByTagName("Főnök");  
 for (int i = 0; i < list.getLength(); i++) {  
 Node node = list.item(i);  
 // Aktuális elem Kiírása  
 System.out.println("\nCurrent element: " + node.getNodeName());  
 if (node.getNodeType() == Node.ELEMENT\_NODE) {  
 Element element = (Element) node;  
 //Fkód eltárolása egy stringbe  
 String fkód=element.getAttribute("fkód");  
 //főnök adatainak kiírása  
 System.out.println("Főnök neve : "  
 + element.getElementsByTagName("Fnév").item(0).getTextContent());  
 System.out.println("Főnök id: " + element.getAttribute("fkód"));  
 System.out.println("Életkora : "  
 +element.getElementsByTagName("Életkor").item(0).getTextContent() );  
 if (node.getNodeType() == Node.ELEMENT\_NODE) {  
 Element elementszemadat = (Element) node;  
 System.out.println("Szem adat : ");  
 System.out.println("Születési helye : " + elementszemadat.getElementsByTagName("Szülhely").item(0).getTextContent());  
 System.out.println("Születési ideje : " + elementszemadat.getElementsByTagName("Szülidő").item(0).getTextContent());  
 }  
 Node nodefkod = list.item(i);  
 if (nodefkod.getNodeType() == Node.ELEMENT\_NODE) {  
 Element element1 = (Element) node;  
 //Fkód vizsgálata és ha megegyezzik 11-el akkor Név módosítása Baranyi Antalra  
 if (fkód.equals("11")) {  
 element1.getElementsByTagName("Fnév").item(0).setTextContent("Baranyi Antal");  
  
 }  
 //főnök új adatainak kiírás  
 System.out.println("\n Új Főnök neve: "  
 + element1.getElementsByTagName("Fnév").item(0).getTextContent());  
 System.out.println("Főnök id: " + element1.getAttribute("fkód"));  
 System.out.println("Életkora : "  
 +element.getElementsByTagName("Életkor").item(0).getTextContent() );  
 if (node.getNodeType() == Node.ELEMENT\_NODE) {  
 Element elementszemadat = (Element) node;  
 System.out.println("Szem adat : ");  
 System.out.println("Születési helye : " + elementszemadat.getElementsByTagName("Szülhely").item(0).getTextContent());  
 System.out.println("Születési ideje : " + elementszemadat.getElementsByTagName("Szülidő").item(0).getTextContent());  
  
 }  
 }  
 }  
 }  
 //Lekérdezi az adatoktak és beleírja egy fájlba  
 Transformer transformer = TransformerFactory.newInstance().newTransformer();  
 Source input=new DOMSource(document);  
 Result output = new StreamResult(new File("XMLcrc7fcLekérdzett.xml"));  
 transformer.transform(input, output);  
  
 } catch (ParserConfigurationException e) {  
 e.printStackTrace();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
}  
}

**Dom query:**

Kilistázza a termékek adatait amelyek tkódja nagyobb mint az 55.

package hu.domparse.crc7fc;  
  
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 javax.xml.xpath.XPath;  
import javax.xml.xpath.XPathConstants;  
import javax.xml.xpath.XPathExpressionException;  
import javax.xml.xpath.XPathFactory;  
import java.io.File;  
import java.io.IOException;  
  
public class DomQuerycrc7fc {  
  
 public static void main(String[] args) {  
 try {  
 DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();  
 DocumentBuilder builder = factory.newDocumentBuilder();  
 //fájl bolvasása  
 Document document = builder.parse(new File("XMLcrc7fc.xml"));  
 document.getDocumentElement().normalize();  
 //Xpath készítése  
 XPath xPath= XPathFactory.newInstance().newXPath();  
 //elérési út megadása  
 //összes termék lekérése  
 // String expression="root/Áruház/Termékek/Termék  
 //csak az 55nél nagyobb id-jű termékek lekérése  
 String expression="root/Áruház/Termékek/Termék[@Tkód>=55]";  
  
 //lista készítés,path kifejezés megírása  
 NodeList nodeList=(NodeList) xPath.compile(expression).evaluate(document, XPathConstants.NODESET);  
 //for ciklus nodlist csomópontjainak végig itrálása  
 for (int i=0;i<nodeList.getLength();i++){  
 Node node=nodeList.item(i);  
 //aktuális elem kiírása  
 System.out.println ("\naktuális elem: " + node.getNodeName());  
  
 //csomópont vizsgálata, találtelemek kilistázása  
 if(node.getNodeType() == Node.ELEMENT\_NODE && node.getNodeName().equals("Termék")){  
 Element element=(Element) node;  
 System.out.println("Áruház id : " + element.getAttribute("Ákód"));  
 System.out.println("Termék id : " + element.getAttribute("Tkód"));  
 System.out.println("Termék neve : " + element.getElementsByTagName("Tnév").item(0).getTextContent());  
 System.out.println("LejáratiDátum : "  
 + element.getElementsByTagName("LejáratiDátum").item(0).getTextContent());  
 System.out.println("Termék ár : " + element.getElementsByTagName("ár").item(0).getTextContent());  
 }  
 }  
 } catch (ParserConfigurationException | XPathExpressionException | IOException | SAXException e){  
 e.printStackTrace();  
 }  
 }  
  
}

**XDM reader:**

Kilistázz minden adatot.

package hu.domparse.crc7fc;  
  
import java.io.File;  
import java.io.IOException;  
  
import javax.xml.parsers.DocumentBuilder;  
import javax.xml.parsers.DocumentBuilderFactory;  
import javax.xml.parsers.ParserConfigurationException;  
  
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;  
  
public class DomReadcrc7fc {  
 public static void main(String[] args) {  
 NodeList list;  
 try {  
 DocumentBuilderFactory documentBuilderFactoryfactory = DocumentBuilderFactory.newInstance();  
 DocumentBuilder documentBuilder = documentBuilderFactoryfactory.newDocumentBuilder();  
  
 // beolvassuk a fájlt  
 Document document = documentBuilder.parse(new File("XMLcrc7fc.xml"));  
 document.getDocumentElement().normalize();  
 //meghatározzuk az aktuális elemet  
 list = document.getElementsByTagName("Főnök");  
 for(int i = 0; i<list.getLength(); i++) {  
 Node node = list.item(i);  
 //aktuális elem kiírása  
 System.out.println("Jelenlegi elem: " + node.getNodeName());  
  
 //Főnök adatainak kiírása  
 if (node.getNodeType() == Node.ELEMENT\_NODE) {  
 Element element = (Element) node;  
 System.out.println("Főnök id : " + element.getAttribute("fkód"));  
 System.out.println("Főnök neve: "  
 + element.getElementsByTagName("Fnév").item(0).getTextContent());  
 System.out.println("Kora : "  
 + element.getElementsByTagName("Életkor").item(0).getTextContent());  
 if (node.getNodeType() == Node.ELEMENT\_NODE) {  
 Element elementszemadat = (Element) node;  
 System.out.println("Szem adat : ");  
 System.out.println("Születési helye : " + elementszemadat.getElementsByTagName("Szülhely").item(0).getTextContent());  
 System.out.println("Születési ideje : " + elementszemadat.getElementsByTagName("Szülidő").item(0).getTextContent());  
  
 }  
  
 }  
  
 }  
 list = document.getElementsByTagName("Áruház");  
 for(int j = 0; j<list.getLength(); j++) {  
 //Áruház adatainak kiírása  
 Node arunode = list.item(j);  
 //aktuális elem kiírása  
 System.out.println("\nJelenlegi elem: " + arunode.getNodeName());  
 if (arunode.getNodeType() == Node.ELEMENT\_NODE) {  
 Element elementaruhaz = (Element) arunode;  
 //áruház adatainak kiírása  
 System.out.println("Áruház id : " + elementaruhaz.getAttribute("Ákód"));  
 System.out.println("Áruház Neve : " + elementaruhaz.getElementsByTagName("Ánév").item(0).getTextContent());  
 System.out.println("Elhelyezkedése : " + elementaruhaz.getElementsByTagName("Hely").item(0).getTextContent());  
 System.out.println("Alapítási éve : " + elementaruhaz.getElementsByTagName("Alapításiév").item(0).getTextContent());  
  
 }  
  
 }  
 list =document.getElementsByTagName("Dolgozók");  
 Node dolgozókknode = list.item(0);  
 //aktuális elem kiírása  
 System.out.println("\nJelenlegi elem: " + dolgozókknode.getNodeName());  
 if (dolgozókknode.getNodeType() == Node.ELEMENT\_NODE) {  
 Element dokelement = (Element) dolgozókknode;  
 //Dolgozók attribútumának kiírása mely a dolgozók fizetése  
 System.out.println("Dolgozói fizetés : " + dokelement.getAttribute("fizetés"));  
 for(int k = 0; k<list.getLength(); k++) {  
 list = document.getElementsByTagName("Dolgozó");  
 Node donode = list.item(k);  
 //aktuális elem kiírása  
 System.out.println("\nJelenlegi elem: " + donode.getNodeName());  
 if (donode.getNodeType() == Node.ELEMENT\_NODE) {  
 Element doelement = (Element) donode;  
 //Dolgozók adatainak kiírása  
 String Dkód = doelement.getAttribute("Dkód");  
 System.out.println("Áruház id : " + doelement.getAttribute("Ákód"));  
 System.out.println("Dolgozó id : " + doelement.getAttribute("Dkód"));  
 System.out.println("Dolgozó neve: "  
 + doelement.getElementsByTagName("Dnév").item(0).getTextContent());  
 //Dolgozó beosztásának kiírása dkód alapján  
 list = document.getElementsByTagName("Beosztás");  
 for(int l = 0; l<list.getLength(); l++) {  
 Node Benode = list.item(l);  
 if (dolgozókknode.getNodeType() == Node.ELEMENT\_NODE) {  
 Element Beelement = (Element) Benode;  
 String dolgozó = Beelement.getAttribute("Dolgozó");  
 //ha a beosztás dolgozók kódja megegyezik a dolgozók dkódjával akkor kiírjuk a napszakát  
 if (Dkód.equals(dolgozó)) {  
 System.out.println("Beosztása : " + Beelement.getElementsByTagName("napszak").item(0).getTextContent());  
 }  
 }  
 }  
 if (donode.getNodeType() == Node.ELEMENT\_NODE) {  
 Element elementdoszemadat = (Element) donode;  
 System.out.println("Szem adat : ");  
 System.out.println("Születési helye : " + elementdoszemadat.getElementsByTagName("Szülhely").item(0).getTextContent());  
 System.out.println("Születési ideje : " + elementdoszemadat.getElementsByTagName("Szülidő").item(0).getTextContent());  
  
 }  
 }  
 }  
 }  
 list =document.getElementsByTagName("Vásárlók");  
 Node Vásáknode = list.item(0);  
 //aktuális elem kiírása  
 System.out.println("\nJelenlegi elem: " + Vásáknode.getNodeName());  
 list = document.getElementsByTagName("Vásárló");  
 for(int k = 0; k<list.getLength(); k++) {  
 Node Vánode = list.item(k);  
 //aktuális elem kiírása  
 System.out.println("\nJelenlegi elem: " + Vánode.getNodeName());  
 if (Vánode.getNodeType() == Node.ELEMENT\_NODE) {  
 Element Váelement = (Element) Vánode;  
 //Vásárlók adatainak kiírása  
 System.out.println("Áruház id : " + Váelement.getAttribute("Ákód"));  
 System.out.println("Vásárló id : " + Váelement.getAttribute("Vkód"));  
 System.out.println("Vásárló neve: " + Váelement.getElementsByTagName("Vnév").item(0).getTextContent());  
 System.out.println("Kora : "  
 + Váelement.getElementsByTagName("Életkor").item(0).getTextContent());  
 if (Vánode.getNodeType() == Node.ELEMENT\_NODE) {  
 Element elementvaszemadat = (Element) Vánode;  
 System.out.println("Szem adat : ");  
 System.out.println("Születési helye : " + elementvaszemadat.getElementsByTagName("Szülhely").item(0).getTextContent());  
 System.out.println("Születési ideje : " + elementvaszemadat.getElementsByTagName("Szülidő").item(0).getTextContent());  
  
 }  
  
 }  
 }  
 list =document.getElementsByTagName("Termékek");  
 Node Terméknode = list.item(0);  
 //aktuális elem kiírása  
 System.out.println("\nJelenlegi elem : " + Terméknode.getNodeName());  
 list = document.getElementsByTagName("Termék");  
 for(int k = 0; k<list.getLength(); k++) {  
 Node tenode = list.item(k);  
 //aktuális elem kiírása  
 System.out.println("\nJelenlegi elem : " + tenode.getNodeName());  
 if (tenode.getNodeType() == Node.ELEMENT\_NODE) {  
 Element Telement = (Element) tenode;  
 //Termékek adatainak kiírása  
 System.out.println("Áruház id : " + Telement.getAttribute("Ákód"));  
 System.out.println("Termék id : " + Telement.getAttribute("Tkód"));  
 System.out.println("Termék neve : " + Telement.getElementsByTagName("Tnév").item(0).getTextContent());  
 System.out.println("LejáratiDátum : "  
 + Telement.getElementsByTagName("LejáratiDátum").item(0).getTextContent());  
 System.out.println("Termék ár : " + Telement.getElementsByTagName("ár").item(0).getTextContent());  
 }  
 }  
 list =document.getElementsByTagName("Termékek");  
 Node menynode = list.item(0);  
 if (dolgozókknode.getNodeType() == Node.ELEMENT\_NODE) {  
 Element menyelement = (Element) menynode;  
 //eladott és vásárolt mennyiség kiírása  
 System.out.println("\nAz eladott mennyiség : " + menyelement.getAttribute("Eladottmennyiség")+" .\nTehát a vásárlók " + menyelement.getAttribute("Eladottmennyiség")+" ásároltak.");  
 }  
  
 } catch (  
 ParserConfigurationException e) {  
 e.printStackTrace();  
 } catch (  
 IOException e) {  
 e.printStackTrace();  
 } catch (  
 SAXException e) {  
 e.printStackTrace();  
 }  
}  
}