JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Emag felépítése

Készítette: **Sándor Máté**

Neptunkód: **AQUSP7**

Dátum: **2022. 11. 28.**

**A feladat leírása:** Egy webáruház, ebben az esetben az EMAG felépítésének XML környezetben való megvalósítása. A szerkezet főbb komponensei az egyedek, gyerekelemei az ezekhez kapcsolódó tulajdonságok, a kapcsolatok pedig az alkotórészek közötti összefüggések. Gyökér elemként az „Emag” elem szolgál. Ennek gyerekelemei a többi elemek (cim, gyarto, termek, tartalmazza, vevo, nev). A „tartalmazza” elem a N:M kapcsolat tulajdonsága alapján lett létrehozva, egy „darabszam” gyerekelemmel. A többértékű tulajdonságok legalább 3 elemmel rendelkeznek a dokumentumban.

**1. Feladat**

**1a) Az adatbázis ER modell**

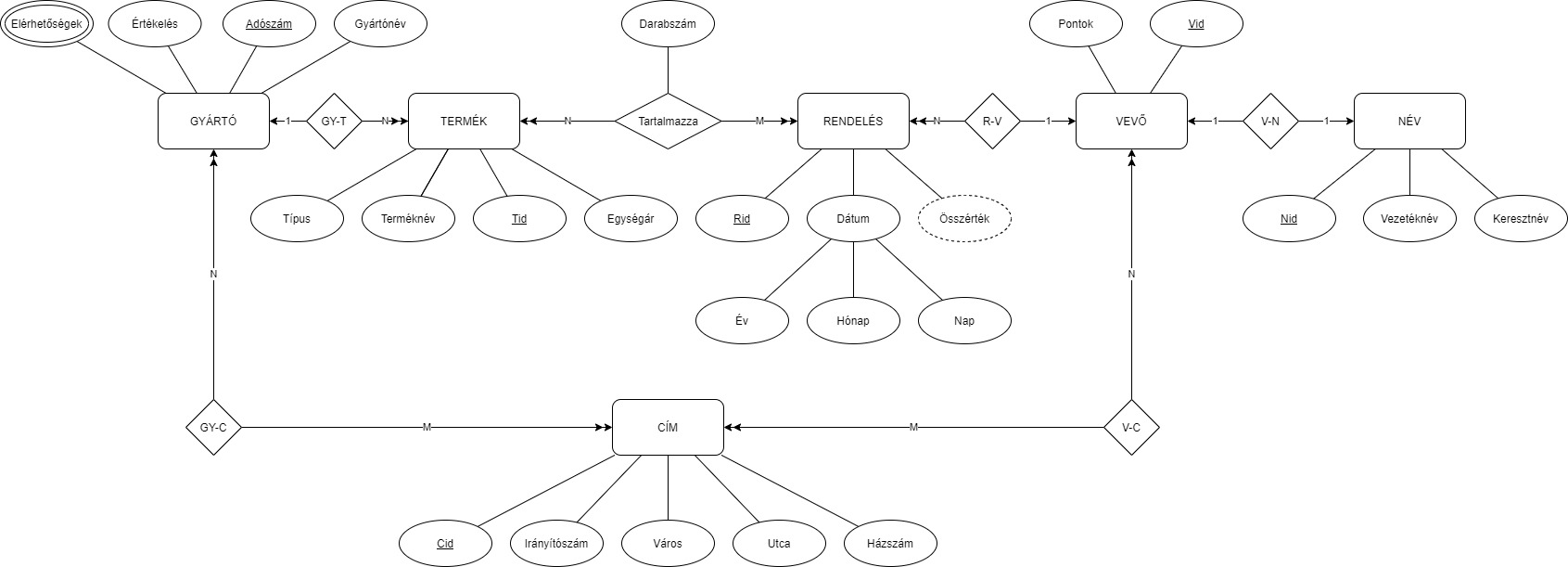
Az **összérték attribútum** származtatott, mivel a a darabszámból és az egységárból számolható.

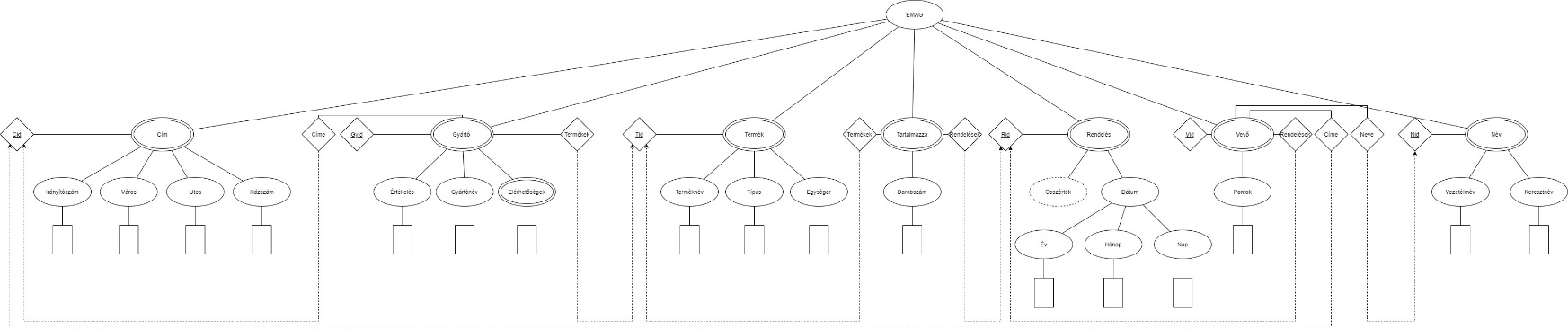
**Cím kapcsolatok:** Egy címhez tartozhat több vevő és több gyártó is, valamint egy gyártónak vagy vevőnek több címe is lehet.

**Tartalmazza kapcsolat:** Egy rendelés több terméket is tartalmazhat, és egy termék több rendelésben is lehet. ezért N-M kapcsolat jön létre amihez létrehoztam a Tartalmazza elemet.

**GY-T kapcsolat:** Egy gyártóhoz több termék is tartozhat, de egy terméket csak egy gyártó szabadalmaztat.

**R-V kapcsolat:** Egy vevőnek több rendelése lehet, de egy rendeléshez csakis egy vevő tartozhat.

**V-N kapcsolat:** Egy vevőnek csak egy neve lehet (vezetéknév, keresztnév)

**1b) Az adatbázis konvertálása XDM modellre**

**1c) Az XDM modell alapján XML dokumentum készítése**

|  |
| --- |
| **<?xml version="1.0" encoding="UTF-8"?>**  **<Emag xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"**  **xsi:noNamespaceSchemaLocation="XMLSchemaAQUSP7.xsd">**  **<!-- Cimek-->**  **<Cim Cid="c1">**  **<iranyitoszam>3720</iranyitoszam>**  **<varos>Sajokaza</varos>**  **<utca>Majus 1 út</utca>**  **<hazszam>11</hazszam>**  **</Cim>**  **<Cim Cid="c2">**  **<iranyitoszam>3720</iranyitoszam>**  **<varos>Sajoivanka</varos>**  **<utca>Petofi Sandor</utca>**  **<hazszam>2</hazszam>**  **</Cim>**  **<Cim Cid="c3">**  **<iranyitoszam>3700</iranyitoszam>**  **<varos>Kazincbarcika</varos>**  **<utca>Majus 1 út</utca>**  **<hazszam>3</hazszam>**  **</Cim>**  **<Cim Cid="c4">**  **<iranyitoszam>3530</iranyitoszam>**  **<varos>Miskolc</varos>**  **<utca>Szent istvan</utca>**  **<hazszam>47</hazszam>**  **</Cim>**  **<Cim Cid="c5">**  **<iranyitoszam>3530</iranyitoszam>**  **<varos>Miskolc</varos>**  **<utca>Joyson utca</utca>**  **<hazszam>1</hazszam>**  **</Cim>**  **<!-- Gyartok-->**  **<Gyarto Gyid="gy1" FTid="t1" FCid="c4">**  **<ertekeles>5</ertekeles>**  **<gyartonev>Makita</gyartonev>**  **<elerhetosegek>Makita.eu</elerhetosegek>**  **</Gyarto>**  **<Gyarto Gyid="gy2" FTid="t2" FCid="c5">**  **<ertekeles>2</ertekeles>**  **<gyartonev>LaptopShop</gyartonev>**  **<elerhetosegek>+36702682881</elerhetosegek>**  **</Gyarto>**  **<Gyarto Gyid="gy3" FTid="t3" FCid="c2">**  **<ertekeles>4</ertekeles>**  **<gyartonev>Ikea</gyartonev>**  **<elerhetosegek>Ikea.com</elerhetosegek>**  **</Gyarto>**  **<!-- Termékek-->**  **<Termek Tid="t1">**  **<termeknev>Elektromos funyíro</termeknev>**  **<tipus>Elektronika</tipus>**  **<egysegar>70000</egysegar>**  **</Termek>**  **<Termek Tid="t2">**  **<termeknev>Thinkpad T450</termeknev>**  **<tipus>Elektronika</tipus>**  **<egysegar>200000</egysegar>**  **</Termek>**  **<Termek Tid="t3">**  **<termeknev>Skorgjorn parna</termeknev>**  **<tipus>Butor</tipus>**  **<egysegar>8000</egysegar>**  **</Termek>**  **<!-- Tartalmazza-->**  **<Tartalmazza Taid="t1" FTid="t1" FRid="r1">**  **<darab>1</darab>**  **</Tartalmazza>**  **<Tartalmazza Taid="t2" FTid="t2" FRid="r2">**  **<darab>2</darab>**  **</Tartalmazza>**  **<Tartalmazza Taid="t3" FTid="t3" FRid="r3">**  **<darab>1</darab>**  **</Tartalmazza>**  **<!-- Rendelesek-->**  **<Rendeles Rid="r1">**  **<osszertek>270000</osszertek>**  **<datum>**  **<ev>2022</ev>**  **<honap>11</honap>**  **<nap>22</nap>**  **</datum>**  **</Rendeles>**  **<Rendeles Rid="r2">**  **<osszertek></osszertek>**  **<datum>**  **<ev>2022</ev>**  **<honap>10</honap>**  **<nap>5</nap>**  **</datum>**  **</Rendeles>**  **<Rendeles Rid="r3">**  **<osszertek></osszertek>**  **<datum>**  **<ev>2021</ev>**  **<honap>12</honap>**  **<nap>21</nap>**  **</datum>**  **</Rendeles>**  **<!-- Vevok-->**  **<Vevo Vid="v1" FNid="n1" FCid="c1" FRid="r1">**  **<pontok>110</pontok>**  **</Vevo>**  **<Vevo Vid="v2" FNid="n2" FCid="c2" FRid="r2">**  **<pontok>0</pontok>**  **</Vevo>**  **<Vevo Vid="v3" FNid ="n3" FCid="c3" FRid="r3">**  **<pontok>10</pontok>**  **</Vevo>**  **<!-- Nevek-->**  **<Nev Nid="n1">**  **<vezeteknev>Sandor</vezeteknev>**  **<keresztnev>Mate</keresztnev>**  **</Nev>**  **<Nev Nid="n2">**  **<vezeteknev>Joska</vezeteknev>**  **<keresztnev>Gyurka</keresztnev>**  **</Nev>**  **<Nev Nid="n3">**  **<vezeteknev>Gabor</vezeteknev>**  **<keresztnev>Aron</keresztnev>**  **</Nev>**  **</Emag>** |

**1d) Az XML dokumentum alapján XMLSchema készítése**

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  <xs:element name="Emag">  <xs:complexType>  <xs:sequence>  <xs:element maxOccurs="unbounded" name="Cim">  <xs:complexType>  <xs:sequence>  <xs:element name="iranyitoszam" type="xs:integer" />  <xs:element name="varos" type="xs:string" />  <xs:element name="utca" type="xs:string" />  <xs:element name="hazszam" type="xs:integer" />  </xs:sequence>  <xs:attribute name="Cid" type="xs:string" use="required" />  </xs:complexType>  </xs:element>  <xs:element maxOccurs="unbounded" name="Gyarto">  <xs:complexType>  <xs:sequence>  <xs:element name="ertekeles" type="xs:integer" />  <xs:element name="gyartonev" type="xs:string" />  <xs:element name="elerhetosegek" type="xs:string" />  </xs:sequence>  <xs:attribute name="Gyid" type="xs:string" use="required" />  <xs:attribute name="FTid" type="xs:string" use="required" />  <xs:attribute name="FCid" type="xs:string" use="required" />  </xs:complexType>  </xs:element>  <xs:element maxOccurs="unbounded" name="Termek">  <xs:complexType>  <xs:sequence>  <xs:element name="termeknev" type="xs:string" />  <xs:element name="tipus" type="xs:string" />  <xs:element name="egysegar" type="xs:integer" />  </xs:sequence>  <xs:attribute name="Tid" type="xs:string" use="required" />  </xs:complexType>  </xs:element>  <xs:element maxOccurs="unbounded" name="Tartalmazza">  <xs:complexType>  <xs:sequence>  <xs:element name="darab" type="xs:integer" />  </xs:sequence>  <xs:attribute name="Taid" type="xs:string" use="required" />  <xs:attribute name="FTid" type="xs:string" use="required" />  <xs:attribute name="FRid" type="xs:string" use="required" />  </xs:complexType>  </xs:element>  <xs:element maxOccurs="unbounded" name="Rendeles">  <xs:complexType>  <xs:sequence>  <xs:element name="osszertek" type="xs:string" nillable="true"/>  <xs:element name="datum">  <xs:complexType>  <xs:sequence>  <xs:element name="ev" type="xs:integer" />  <xs:element name="honap" type="xs:integer" />  <xs:element name="nap" type="xs:integer" />  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  <xs:attribute name="Rid" type="xs:string" use="required" />  </xs:complexType>  </xs:element>  <xs:element maxOccurs="unbounded" name="Vevo">  <xs:complexType>  <xs:sequence>  <xs:element name="pontok" type="xs:integer" />  </xs:sequence>  <xs:attribute name="Vid" type="xs:string" use="required" />  <xs:attribute name="FNid" type="xs:string" use="required" />  <xs:attribute name="FCid" type="xs:string" use="required" />  <xs:attribute name="FRid" type="xs:string" use="required" />  </xs:complexType>  </xs:element>  <xs:element maxOccurs="unbounded" name="Nev">  <xs:complexType>  <xs:sequence>  <xs:element name="vezeteknev" type="xs:string" />  <xs:element name="keresztnev" type="xs:string" />  </xs:sequence>  <xs:attribute name="Nid" type="xs:string" use="required" />  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>    <!-- Kulcsok-->  <xs:key name="Cid">  <xs:selector xpath="Cim" />  <xs:field xpath="@Cid" />  </xs:key>  <xs:key name="Gyid">  <xs:selector xpath="Gyarto" />  <xs:field xpath="@Gyid" />  </xs:key>  <xs:key name="Tid">  <xs:selector xpath="Termek" />  <xs:field xpath="@nev" />  </xs:key>  <xs:key name="Taid">  <xs:selector xpath="Tartalmazza" />  <xs:field xpath="@Taid" />  </xs:key>  <xs:key name="Vid">  <xs:selector xpath="Vevo" />  <xs:field xpath="@Vid" />  </xs:key>  <xs:key name="Nid">  <xs:selector xpath="Nev" />  <xs:field xpath="@Nid" />  </xs:key>    <!-- Idegen kulcsok -->  <xs:keyref refer="Cid" name="Cim\_idegen\_kulcs">  <xs:selector xpath="Rendeles" />  <xs:field xpath="@FCid" />  </xs:keyref>  <xs:keyref refer="Gyid" name="Gyarto\_idegen\_kulcs">  <xs:selector xpath="Rendeles" />  <xs:field xpath="@FGyid" />  </xs:keyref>  <xs:keyref refer="Tid" name="Termek\_Cim\_idegen\_kulcs">  <xs:selector xpath="Cim" />  <xs:field xpath="@FTid" />  </xs:keyref>  <xs:keyref refer="Cid" name="Cim\_Tartalmazza\_idegen\_kulcs">  <xs:selector xpath="Tartalmazza" />  <xs:field xpath="@FCid" />  </xs:keyref>  <xs:keyref refer="Gyid" name="Gyarto\_Tartalmazza\_idegen\_kulcs">  <xs:selector xpath="Tartalmazza" />  <xs:field xpath="@FGyid" />  </xs:keyref>    </xs:element>  </xs:schema> |

**2. feladat – DOM program készítés**

**A DOM programok Eclipse-ben készültek. A kódban sok commentet hagytam az egyszerű értelmezésért, a maradék amit nem commenteztem az triviális a gyakorlatban készült programok alapján.**

**2a) Adatolvasás, kiírás konzolra – fájlba**

|  |
| --- |
| package hu.domparse.aqusp7;  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 DomReadAQUSP7 {  public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException {  //Forrás file  File file = new File("XMLAQUSP7.xml");    DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();  DocumentBuilder dBuilder = factory.newDocumentBuilder();    Document doc = dBuilder.parse(file);    doc.getDocumentElement().normalize();  //Gyökérelem  System.out.println("Root element: " + doc.getDocumentElement().getNodeName());  //Gyerekelemek lementése  NodeList nList = (NodeList) doc.getDocumentElement();      for (int i = 0; i < nList.getLength(); i++) {  Node node = nList.item(i);    //Ha Cim  if(node.getNodeName() == "Cim"){  if(!node.getNodeName().equals("#text")) {  System.out.println("\n");  System.out.println("Current element: " + node.getNodeName());  }  //Cím adatainak kiírása  if(node.getNodeType()==Node.ELEMENT\_NODE) {  Element elem = (Element) node;    String cid = elem.getAttribute("Cid");    Node cim\_node = elem.getElementsByTagName("iranyitoszam").item(0);  String iranyitoszam = cim\_node.getTextContent();    Node cim\_node2 = elem.getElementsByTagName("varos").item(0);  String varos = cim\_node2.getTextContent();    Node cim\_node3 = elem.getElementsByTagName("utca").item(0);  String utca = cim\_node3.getTextContent();    Node cim\_node4 = elem.getElementsByTagName("hazszam").item(0);  String hazszam = cim\_node4.getTextContent();    System.out.printf("Cid: %s%n", cid);  System.out.printf("Iranyitoszam: %s%n", iranyitoszam);  System.out.printf("Varos: %s%n", varos);  System.out.printf("Utca: %s%n", utca);  System.out.printf("Hazszam: %s%n", hazszam);  }  }      }  }  } |

**2b) Adatmódosítás**

|  |
| --- |
| **package hu.domparse.aqusp7;**  **import java.io.File;**  **import javax.xml.parsers.DocumentBuilderFactory;**  **import javax.xml.parsers.DocumentBuilder;**  **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.NodeList;**  **import org.w3c.dom.Node;**  **import org.w3c.dom.Element;**  **import org.w3c.dom.NamedNodeMap;**  **public class DomModifyAQUSP7**  **{**  **public static void main(String argv[])**  **{**  **try**  **{**  **File inputFile = new File("XMLAQUSP7.xml");**  **DocumentBuilderFactory docFactory = DocumentBuilderFactory.newInstance();**  **DocumentBuilder docBuilder = docFactory.newDocumentBuilder();**  **Document doc = docBuilder.parse(inputFile);**    **// ------------------------------------------------------------------------------------------------------------------------------**  **//Attribútum értékeket változtat, feltétellel for loopban az összes címre.**    **NodeList cimList = doc.getElementsByTagName("Cim");**  **for(int i = 0; i < cimList.getLength(); i++)**  **{**  **Node cim = doc.getElementsByTagName("Cim").item(i);**  **NamedNodeMap attr = cim.getAttributes();**  **Node nodeAttr = attr.getNamedItem("Cid");**  **nodeAttr.setTextContent("lakcim" + (i+1));**    **NodeList list = cim.getChildNodes();**    **// for loop ami a gyerekelemek számáig megy**  **for(int temp = 0; temp < list.getLength(); temp++)**  **{**  **Node node = list.item(temp);**  **if (node.getNodeType() == Node.ELEMENT\_NODE)**  **{**  **Element eElement = (Element) node;**  **// A varos nevű gyerekelemnél teljesül**  **if ("varos".equals(eElement.getNodeName()))**  **{**  **// Ha a varos egyenlő ezzel**  **if ("Miskolc".equals(eElement.getTextContent()))**  **{**  **// Változtassa meg erre**  **eElement.setTextContent("Szirma");**  **}**  **}**  **}**  **}**  **}**    **// ------------------------------------------------------------------------------------------------------------------------------**    **//Gyerekelem törlés**    **NodeList gyartoList = doc.getElementsByTagName("Gyarto");**  **for(int i = 0; i < gyartoList.getLength(); i++)**  **{**  **// Kilistázza a jelenlegi gyarto egyedet**  **Node gyarto = doc.getElementsByTagName("Gyarto").item(i);**    **// Lekéri annak gyerekelemeit**  **NodeList childNodes = gyarto.getChildNodes();**  **// Végigmegy a gyerekelemeken**  **for (int count = 0; count < childNodes.getLength(); count++)**  **{**  **Node node = childNodes.item(count);**  **// Ha a gyerekelem neve "ertekeles"**  **if("ertekeles".equals(node.getNodeName()))**  **{**  **// Akkor törölje**  **gyarto.removeChild(node);**  **}**  **}**  **}**    **// ------------------------------------------------------------------------------------------------------------------------------**    **// A dátumon belül módosítja az évbe írtakat**    **NodeList datumList = doc.getElementsByTagName("datum");**  **for(int i = 0; i < datumList.getLength(); i++)**  **{**  **// Kilistázza a datum egyedeket**  **Node datum = doc.getElementsByTagName("datum").item(i);**    **// Lekéri annak gyerekelemeit**  **NodeList childNodes = datum.getChildNodes();**  **// for loop ami a gyerekelemek számáig megy**  **for(int temp = 0; temp < childNodes.getLength(); temp++)**  **{**  **Node node = childNodes.item(temp);**  **// Ellenőrzés hogy a kapott egyed elem-e**  **if(node.getNodeType() == Node.ELEMENT\_NODE)**  **{**  **Element eElement = (Element) node;**  **// Az ev nevű gyerekelemnél teljesül**  **if("ev".equals(eElement.getNodeName()))**  **{**  **// Ha az év egyenlő ezzel**  **if ("2022".equals(eElement.getTextContent()))**  **{**  **// Változtassa meg "now"-ra**  **eElement.setTextContent("now");**  **}**  **}**  **}**  **}**  **}**  **// ------------------------------------------------------------------------------------------------------------------------------**  **// Új gyerekelemet vesz fel a Tartalmazza egyedbe - "kupon", majd az FRid attribútum értéke alapján állít neki értéket**    **NodeList tartalmazzaList = doc.getElementsByTagName("Tartalmazza");**  **for (int i = 0; i < tartalmazzaList.getLength(); i++)**  **{**  **Node tartalmazza = tartalmazzaList.item(i);**  **// Lekéri az "FRid" attribútum értékét és eltárolja az "id"-ben**  **String id = tartalmazza.getAttributes().getNamedItem("FRid").getTextContent();**  **// Létrehozza az új "kupon" elemet**  **Element kupon = doc.createElement("kupon");**  **tartalmazza.appendChild(kupon);**  **// Az "id" értéke alapján ad értéket az új "kupon" elemnek**  **if ("r1".equals(id))**  **{**  **kupon.appendChild(doc.createTextNode("0"));**  **}**  **if ("r2".equals(id))**  **{**  **kupon.appendChild(doc.createTextNode("30"));**  **}**  **if ("r3".equals(id))**  **{**  **kupon.appendChild(doc.createTextNode("0"));**  **}**  **}**  **// ------------------------------------------------------------------------------------------------------------------------------**  **// Tratalom konzolra írása:**    **TransformerFactory transformerFactory = TransformerFactory.newInstance();**  **Transformer transformer = transformerFactory.newTransformer();**  **DOMSource source = new DOMSource(doc);**  **System.out.println("-----Módosított fájl-----");**  **StreamResult consoleResult = new StreamResult(System.out);**  **transformer.transform(source, consoleResult);**  **} catch (Exception e)**  **{**  **e.printStackTrace();**  **}**  **}**  **}** |

**2c) Adatlekérdezés**

|  |
| --- |
| package hu.domparse.aqusp7;  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 DomQueryAQUSP7  {  public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException  {  //Forrás file  File file = new File("XMLAQUSP7.xml");  DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();  DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();    Document doc = dBuilder.parse(file);  doc.getDocumentElement().normalize();  // Gyökér elem  System.out.print("Gyökér elem: ");  System.out.println(doc.getDocumentElement().getNodeName());  // Címek kilistázása  NodeList nList = doc.getElementsByTagName("Cim");    System.out.println("-----------------");    // Végigfut az "Cim"-nek a gyerekelemein, kihagyva az "iranyitoszamot"-t  for(int i = 0; i < nList.getLength(); i++)  {  Node node = nList.item(i);  System.out.println("\nJelenlegi elem: " + node.getNodeName());  if(node.getNodeType() == Node.ELEMENT\_NODE)  {  Element elem = (Element) node;  System.out.println("Cid: " + elem.getAttribute("Cid"));  NodeList nList2 = elem.getChildNodes();  for(int j = 0; j < nList2.getLength(); j++)  {  Node node2 = nList2.item(j);  if(node2.getNodeType() == Node.ELEMENT\_NODE)  {  if(!node2.getNodeName().equals("iranyitoszam"))  {  System.out.println(node2.getNodeName() + " : " + node2.getTextContent());  }  }  }  }  }  }  } |