JEGYZŐKÖNYV

Adatkezelés XML környezetben Féléves feladat

Készítette: **Nagy Máté**

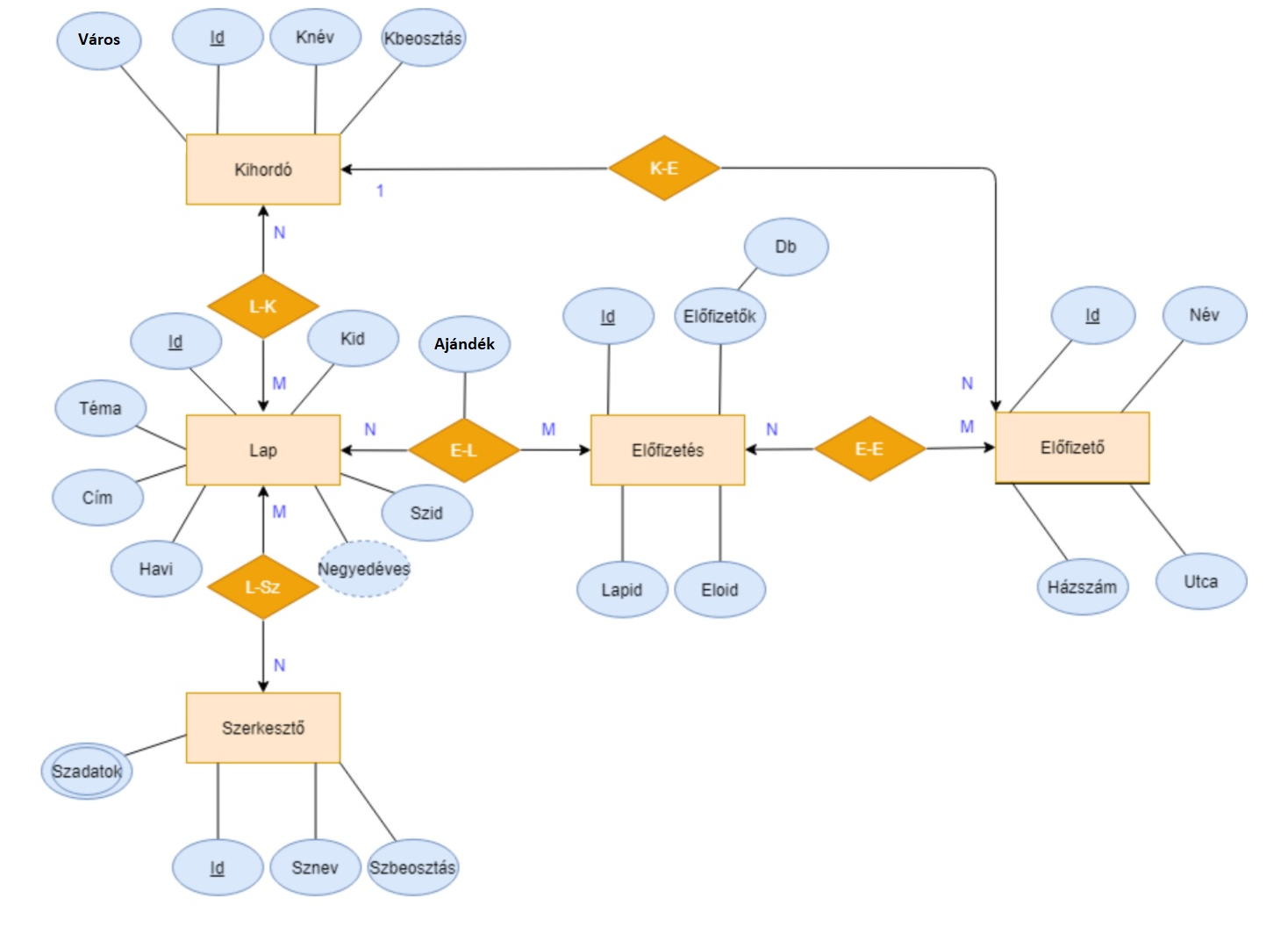
Neptunkód: **CWM4U0**

**A feladat leírása:**

Feladatom egy olyan adatbázis ER modelljének a megtervezése, ami megfelel a normálformáknak és nem tartalmaz redundanciát. Az adatbázisom lapok és a hozzájuk köthető személyek adataiból fog állni pl kihordó vagy előfizető és ezeken végzem el szükséges feladatokat. Itt készített leírás tartalmazni fogja a kódokat pl xml vagy xds valamint programmal rajzolt ábrákat is.

**1.Feladat**

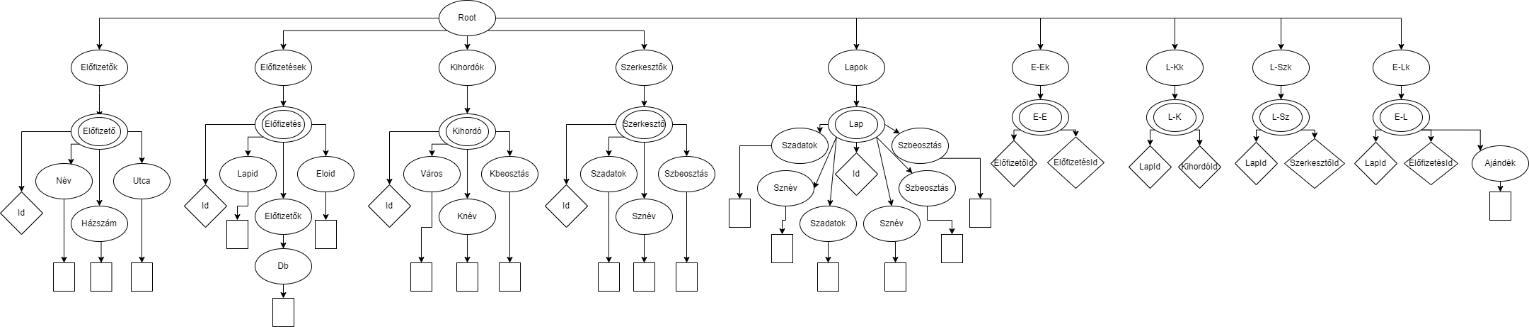
# 1a) Az adatbázis ER modell



# 1b) Az adatbázis ER modell

XDM modellben az elemet ellipszissel ábrázoljuk.

Az attribútumot rombusszal lehet jelölni, illetve a szöveget a téglalapban. Dupla ellipszis, amely a többszörös előfordulás. Minden N:M kapcsolathoz felvesszük az egyed többesszám alakját is.



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

<?xml version="1.0" encoding="UTF-8"?>  
<root xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="XMLSchemaCWM4U0.xsd">  
<Elofizetok>  
 <Elofizeto ElofizetoId="1" ElofizetesId="1">  
 <Nev>Gábor</Nev>  
 <Utca>Eper</Utca>  
 <Hazszam>1</Hazszam>  
 </Elofizeto>  
 <Elofizeto ElofizetoId="2" ElofizetesId="2">  
 <Nev>Bence</Nev>  
 <Utca>Eper</Utca>  
 <Hazszam>2</Hazszam>  
 </Elofizeto>  
 <Elofizeto ElofizetoId="3" ElofizetesId="3">  
 <Nev>Lajos</Nev>  
 <Utca>Eper</Utca>  
 <Hazszam>3</Hazszam>  
 </Elofizeto>  
</Elofizetok>  
<Elofizetesek>  
 <Elofizetes ElofizetesId="1">  
 <Lapid>1</Lapid>  
 <Eloid>1</Eloid>  
 <Elofizetok>  
 <Db>1</Db>  
 </Elofizetok>  
 </Elofizetes>  
 <Elofizetes ElofizetesId="2">  
 <Lapid>2</Lapid>  
 <Eloid>2</Eloid>  
 <Elofizetok>  
 <Db>2</Db>  
 </Elofizetok>  
 </Elofizetes>  
 <Elofizetes ElofizetesId="3">  
 <Lapid>3</Lapid>  
 <Eloid>3</Eloid>  
 <Elofizetok>  
 <Db>3</Db>  
 </Elofizetok>  
 </Elofizetes>  
</Elofizetesek>  
<Kihordok>  
 <Kihordo Kihordoid="1">  
 <Knev>Endre</Knev>  
 <Kbeosztas>1</Kbeosztas>  
 <Varos>Miskolc</Varos>  
 </Kihordo>  
 <Kihordo Kihordoid="2">  
 <Knev>Lajos</Knev>  
 <Kbeosztas>2</Kbeosztas>  
 <Varos>Budapest</Varos>  
 </Kihordo>  
 <Kihordo Kihordoid="3">  
 <Knev>Milán</Knev>  
 <Kbeosztas>3</Kbeosztas>  
 <Varos>Miskolc</Varos>  
 </Kihordo>  
</Kihordok>  
<Szerkesztok>  
 <Szerkeszto SzerkesztoId="1">  
 <Sznev>Imre</Sznev>  
 <Szbeosztas>Sport elemzo</Szbeosztas>  
 <Szadatok>Egyetemi végzettség</Szadatok>  
 </Szerkeszto>  
 <Szerkeszto SzerkesztoId="2">  
 <Sznev>Ferenc</Sznev>  
 <Szbeosztas>Riporter</Szbeosztas>  
 <Szadatok>Középiskolai végzettség</Szadatok>  
 </Szerkeszto>  
 <Szerkeszto SzerkesztoId="3">  
 <Sznev>Erik</Sznev>  
 <Szbeosztas>Designer</Szbeosztas>  
 <Szadatok>Egyetemi végzettség</Szadatok>  
 </Szerkeszto>  
</Szerkesztok>  
<Lapok>  
 <Lap LapId="1">  
 <Cim>Nemzeti sport</Cim>  
 <Tema>Sport</Tema>  
 <Havi>6000</Havi>  
 <Negyedeves>17850</Negyedeves>  
 <Szid>1</Szid>  
 <Kid>1</Kid>  
 </Lap>  
 <Lap LapId="2">  
 <Cim>PC World</Cim>  
 <Tema>Informatika</Tema>  
 <Havi>1800</Havi>  
 <Negyedeves>5385</Negyedeves>  
 <Szid>2</Szid>  
 <Kid>2</Kid>  
 </Lap>  
 <Lap LapId="3">  
 <Cim>Mi Újság</Cim>  
 <Tema>Hírek</Tema>  
 <Havi>1500</Havi>  
 <Negyedeves>5000</Negyedeves>  
 <Szid>3</Szid>  
 <Kid>3</Kid>  
 </Lap>  
</Lapok>  
<E-Ek>  
 <E-E ElofizetoId="1" ElofizetesId="1">  
 </E-E>  
 <E-E ElofizetoId="2" ElofizetesId="2">  
 </E-E>  
 <E-E ElofizetoId="3" ElofizetesId="3">  
 </E-E>  
</E-Ek>  
<L-Kk>  
 <L-K LapId="1" Kihordoid="1">  
 </L-K>  
 <L-K LapId="2" Kihordoid="2">  
 </L-K>  
 <L-K LapId="3" Kihordoid="3">  
 </L-K>  
</L-Kk>  
<L-Szk>  
 <L-Sz LapId="1" SzerkesztoId="1">  
 </L-Sz>  
 <L-Sz LapId="2" SzerkesztoId="2">  
 </L-Sz>  
 <L-Sz LapId="3" SzerkesztoId="3">  
 </L-Sz>  
</L-Szk>  
<E-Lk>  
 <E-L LapId="1" ElofizetesId="1">  
 <Ajandek>Nincs</Ajandek>  
 </E-L>  
 <E-L LapId="2" ElofizetesId="2">  
 <Ajandek>Van</Ajandek>  
 </E-L>  
 <E-L LapId="3" ElofizetesId="3">  
 <Ajandek>Nincs</Ajandek>  
 </E-L>  
</E-Lk>  
</root>

# 1d) Az XML dokumentum alapján XMLSchema készítése

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  
 <xs:element name="root">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="Elofizetok">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="Elofizeto" maxOccurs="unbounded" minOccurs="0">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element type="xs:string" name="Nev"/>  
 <xs:element type="xs:string" name="Utca"/>  
 <xs:element type="xs:byte" name="Hazszam"/>  
 </xs:sequence>  
 <xs:attribute type="xs:byte" name="ElofizetoId" use="optional"/>  
 <xs:attribute type="xs:byte" name="ElofizetesId" use="optional"/>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 <xs:element name="Elofizetesek">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="Elofizetes" maxOccurs="unbounded" minOccurs="0">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element type="xs:byte" name="Lapid"/>  
 <xs:element type="xs:byte" name="Eloid"/>  
 <xs:element name="Elofizetok">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element type="xs:byte" name="Db"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 <xs:attribute type="xs:byte" name="ElofizetesId" use="optional"/>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 <xs:element name="Kihordok">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="Kihordo" maxOccurs="unbounded" minOccurs="0">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element type="xs:string" name="Knev"/>  
 <xs:element type="xs:byte" name="Kbeosztas"/>  
 <xs:element type="xs:string" name="Varos"/>  
 </xs:sequence>  
 <xs:attribute type="xs:byte" name="Kihordoid" use="optional"/>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 <xs:element name="Szerkesztok">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="Szerkeszto" maxOccurs="unbounded" minOccurs="0">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element type="xs:string" name="Sznev"/>  
 <xs:element type="xs:string" name="Szbeosztas"/>  
 <xs:element type="xs:string" name="Szadatok"/>  
 </xs:sequence>  
 <xs:attribute type="xs:byte" name="SzerkesztoId" use="optional"/>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 <xs:element name="Lapok">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="Lap" maxOccurs="unbounded" minOccurs="0">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element type="xs:string" name="Cim"/>  
 <xs:element type="xs:string" name="Tema"/>  
 <xs:element type="ar" name="Havi"/>  
 <xs:element type="ar" name="Negyedeves"/>  
 <xs:element type="xs:byte" name="Szid"/>  
 <xs:element type="xs:byte" name="Kid"/>  
 </xs:sequence>  
 <xs:attribute type="xs:byte" name="LapId" use="optional"/>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 <xs:element name="E-Ek">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="E-E" maxOccurs="unbounded" minOccurs="0">  
 <xs:complexType>  
 <xs:simpleContent>  
 <xs:extension base="xs:string">  
 <xs:attribute type="xs:byte" name="ElofizetoId" use="optional"/>  
 <xs:attribute type="xs:byte" name="ElofizetesId" use="optional"/>  
 </xs:extension>  
 </xs:simpleContent>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 <xs:element name="L-Kk">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="L-K" maxOccurs="unbounded" minOccurs="0">  
 <xs:complexType>  
 <xs:simpleContent>  
 <xs:extension base="xs:string">  
 <xs:attribute type="xs:byte" name="LapId" use="optional"/>  
 <xs:attribute type="xs:byte" name="Kihordoid" use="optional"/>  
 </xs:extension>  
 </xs:simpleContent>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 <xs:element name="L-Szk">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="L-Sz" maxOccurs="unbounded" minOccurs="0">  
 <xs:complexType>  
 <xs:simpleContent>  
 <xs:extension base="xs:string">  
 <xs:attribute type="xs:byte" name="LapId" use="optional"/>  
 <xs:attribute type="xs:byte" name="SzerkesztoId" use="optional"/>  
 </xs:extension>  
 </xs:simpleContent>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 <xs:element name="E-Lk">  
 <xs:complexType>  
 <xs:choice maxOccurs="unbounded" minOccurs="0">  
 <xs:element name="E-L">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element type="xs:string" name="Ajandek"/>  
 </xs:sequence>  
 <xs:attribute type="xs:byte" name="LapId" use="optional"/>  
 <xs:attribute type="xs:byte" name="ElofizetesId" use="optional"/>  
 </xs:complexType>  
 </xs:element>  
 </xs:choice>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 </xs:complexType>  
 <xs:key name="Kihordo\_Key">  
 <xs:selector xpath="Kihordo"/>  
 <xs:field xpath="@DId"/>  
 </xs:key>  
 <xs:key name="Elofizeto\_Key">  
 <xs:selector xpath="Elofizeto"/>  
 <xs:field xpath="@DId"/>  
 </xs:key>  
 <xs:key name="Elofizetes\_Key">  
 <xs:selector xpath="Elofizetes"/>  
 <xs:field xpath="@DId"/>  
 </xs:key>  
 <xs:key name="Lap\_Key">  
 <xs:selector xpath="Lap"/>  
 <xs:field xpath="@DId"/>  
 </xs:key>  
 <xs:key name="Szerkeszto\_Key">  
 <xs:selector xpath="Szerkeszto"/>  
 <xs:field xpath="@DId"/>  
 </xs:key>  
 <xs:keyref name="Fkey\_Lap\_Kihordo" refer="Kihordo\_Key">  
 <xs:selector xpath="Lap"/>  
 <xs:field xpath="@Id"/>  
 </xs:keyref>  
 <xs:keyref name="Fkey\_Elofizetes\_Elofizeto" refer="Elofizeto\_Key">  
 <xs:selector xpath="Elofizetes"/>  
 <xs:field xpath="@Id"/>  
 </xs:keyref>  
 <xs:keyref name="Fkey\_Lap\_Szerkeszto" refer="Szerkeszto\_Key">  
 <xs:selector xpath="Lap"/>  
 <xs:field xpath="@Id"/>  
 </xs:keyref>  
 <xs:keyref name="Fkey\_Elofizetes\_Lap" refer="Elofizetes\_Key">  
 <xs:selector xpath="Lap"/>  
 <xs:field xpath="@Id"/>  
 </xs:keyref>  
 </xs:element>  
 <xs:simpleType name="ar">  
 <xs:restriction base="xs:integer">  
 <xs:minInclusive value="1"/>  
 <xs:maxInclusive value="100000"/>  
 </xs:restriction>  
 </xs:simpleType>  
</xs:schema>

**2. feladat**

## 2a) Adatolvasás

package hu.domparse.cwm4u0;  
  
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 DomReadCWM4U0 {  
 public static void main(String[] args) {  
 NodeList list;  
 try {  
 DocumentBuilderFactory documentBuilderFactoryfactory = DocumentBuilderFactory.newInstance();  
 DocumentBuilder documentBuilder = documentBuilderFactoryfactory.newDocumentBuilder();  
  
 // Fájl beolvasás  
 Document document = documentBuilder.parse(new File("XMLcwm4u0.xml"));  
 document.getDocumentElement().normalize();  
 //Elem meghatározás  
 list = document.getElementsByTagName("Elofizeto");  
 for(int i = *0*; i<list.getLength(); i++) {  
 Node node = list.item(i);  
 //Elem kiírása  
 System.out.println("*\n*Aktuális elem: " + node.getNodeName());  
  
 //Előfizető adatainak kiírása  
 if (node.getNodeType() == Node.ELEMENT\_NODE) {  
 Element element = (Element) node;  
 System.out.println("Előfizető neve : " + element.getElementsByTagName("Nev").item(*0*).getTextContent());  
 System.out.println("Előfizető utca: "  
 + element.getElementsByTagName("Utca").item(*0*).getTextContent());  
 System.out.println("Hazszam : "  
 + element.getElementsByTagName("Hazszam").item(*0*).getTextContent());  
 }  
  
 }  
 list = document.getElementsByTagName("Kihordo");  
 for(int j = *0*; j<list.getLength(); j++) {  
 //Kihordó adatainak kiírása  
 Node kihordonode = list.item(j);  
 //Elem kiírása  
 System.out.println("*\n*Aktuális elem: " + kihordonode.getNodeName());  
 if (kihordonode.getNodeType() == Node.ELEMENT\_NODE) {  
 Element elementkihordo = (Element) kihordonode;  
 //Előfizetes adatainak kiírása  
 System.out.println("Név : " + elementkihordo.getElementsByTagName("Knev").item(*0*).getTextContent());  
 System.out.println("Beosztás : " + elementkihordo.getElementsByTagName("Kbeosztas").item(*0*).getTextContent());  
 System.out.println("Város : " + elementkihordo.getElementsByTagName("Varos").item(*0*).getTextContent());  
 }  
 }  
 list = document.getElementsByTagName("Elofizetes");  
 for(int j = *0*; j<list.getLength(); j++) {  
 //Előfizetes adatainak kiírása  
 Node elonode = list.item(j);  
 //Elem kiírása  
 System.out.println("*\n*Aktuális elem: " + elonode.getNodeName());  
 if (elonode.getNodeType() == Node.ELEMENT\_NODE) {  
 Element elementelofizetes = (Element) elonode;  
 //Előfizetes adatainak kiírása  
 System.out.println("Lap id : " + elementelofizetes.getAttribute("Lapid"));  
 System.out.println("Elo id : " + elementelofizetes.getAttribute("Eloid"));  
 }  
 if (elonode.getNodeType() == Node.ELEMENT\_NODE) {  
 Element elementdb = (Element) elonode;  
 System.out.println("Darab : " + elementdb.getElementsByTagName("Db").item(*0*).getTextContent());  
 }  
 }  
 list = document.getElementsByTagName("Szerkeszto");  
 for(int j = *0*; j<list.getLength(); j++) {  
 //Előfizetes adatainak kiírása  
 Node szerknode = list.item(j);  
 //Elem kiírása  
 System.out.println("*\n*Aktuális elem: " + szerknode.getNodeName());  
 if (szerknode.getNodeType() == Node.ELEMENT\_NODE) {  
 Element elementelofizetes = (Element) szerknode;  
 //Előfizetes adatainak kiírása  
 System.out.println("Szerkesztő neve : " + elementelofizetes.getElementsByTagName("Sznev").item(*0*).getTextContent());  
 System.out.println("Szerkesztő beosztas : " + elementelofizetes.getElementsByTagName("Szbeosztas").item(*0*).getTextContent());  
 System.out.println("Szerkeszto adatai : " + elementelofizetes.getElementsByTagName("Szadatok").item(*0*).getTextContent());  
 }  
 }  
  
 list = document.getElementsByTagName("Lap");  
 for(int j = *0*; j<list.getLength(); j++) {  
 //Lap adatainak kiírása  
 Node lapnode = list.item(j);  
 //Elem kiírása  
 System.out.println("*\n*Aktuális elem: " + lapnode.getNodeName());  
 if (lapnode.getNodeType() == Node.ELEMENT\_NODE) {  
 Element elementelofizetes = (Element) lapnode;  
 //Előfizetes adatainak kiírása  
 System.out.println("Lap címe : " + elementelofizetes.getElementsByTagName("Cim").item(*0*).getTextContent());  
 System.out.println("Lap téma : " + elementelofizetes.getElementsByTagName("Tema").item(*0*).getTextContent());  
 System.out.println("Lap havi : " + elementelofizetes.getElementsByTagName("Havi").item(*0*).getTextContent());  
 System.out.println("Lap negyedéves : " + elementelofizetes.getElementsByTagName("Negyedeves").item(*0*).getTextContent());  
 System.out.println("Lap szid : " + elementelofizetes.getElementsByTagName("Szid").item(*0*).getTextContent());  
 System.out.println("Lap kid : " + elementelofizetes.getElementsByTagName("Kid").item(*0*).getTextContent());  
  
 }  
 }  
  
  
 } catch (  
 ParserConfigurationException e) {  
 e.printStackTrace();  
 } catch (  
 IOException e) {  
 e.printStackTrace();  
 } catch (  
 SAXException e) {  
 e.printStackTrace();  
 }  
 }  
}

## 2b) Adatmódositás

package hu.domparse.cwm4u0;  
  
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 DomModifyCWM4U0 {  
 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("XMLcwm4u0.xml"));  
 document.getDocumentElement().normalize();  
 // Az elem meghatározása  
 list = document.getElementsByTagName("Elofizeto");  
 for (int i = 0; i < list.getLength(); i++) {  
 Node node = list.item(i);  
 // Az elem kiírása  
 System.*out*.println("\nAktuális elem: " + node.getNodeName());  
 if (node.getNodeType() == Node.*ELEMENT\_NODE*) {  
 Element element = (Element) node;  
 //Id eltárolása egy stringbe  
 String id=element.getAttribute("ElofizetoId");  
 //Elofizetők adatainak kiírása  
 System.*out*.println("Előfizető id : "  
 + element.getAttribute("ElofizetoId"));  
 System.*out*.println("Előfizetők neve : "  
 + element.getElementsByTagName("Nev").item(0).getTextContent());  
 System.*out*.println("Előfizetők utca: " + element.getElementsByTagName("Utca").item(0).getTextContent());  
 System.*out*.println("Előfizetők haszszam: "  
 +element.getElementsByTagName("Hazszam").item(0).getTextContent() );  
 Node nodenev = list.item(i);  
 if (nodenev.getNodeType() == Node.*ELEMENT\_NODE*) {  
 Element element1 = (Element) node;  
 //Id vizsgálata és ha megegyezzik 1-el akkor Név módosítása Bélára  
 if (id.equals("1")) {  
 System.*out*.println("Siker");  
 element1.getElementsByTagName("Nev").item(0).setTextContent("Béla");  
  
 }  
 //Elofizetők új adatainak kiírás  
 System.*out*.println("Előfizetők neve : "  
 + element.getElementsByTagName("Nev").item(0).getTextContent());  
 System.*out*.println("Előfizetők utca: " + element.getElementsByTagName("Utca").item(0).getTextContent());  
 System.*out*.println("Előfizetők haszszam: "  
 +element.getElementsByTagName("Hazszam").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("XMLcwm4u0mod.xml"));  
 transformer.transform(input, output);  
  
 } catch (ParserConfigurationException e) {  
 e.printStackTrace();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 }  
}

# 3c) Adatlekérdezés

package hu.domparse.cwm4u0;  
  
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 DomQueryCWM4U0 {  
  
 public static void main(String[] args) {  
 try {  
 DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();  
 DocumentBuilder builder = factory.newDocumentBuilder();  
 //Fájl bolvasása  
 Document document = builder.parse(new File("XMLcwm4u0.xml"));  
 document.getDocumentElement().normalize();  
 //Xpath készítése  
 XPath xPath= XPathFactory.newInstance().newXPath();  
 //Elérési út megadása  
 //Összes előfizető lekérése  
 String expression="root/Elofizetok/Elofizeto";  
  
 //Lista elkészítése,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);  
 //Elem kiírása  
 System.out.println ("*\n*Aktuális elem: " + node.getNodeName());  
  
 //Vizsgálata, találtelemek kilistázása  
 if(node.getNodeName().equals("Elofizeto")){  
 Element element=(Element) node;  
 System.out.println("Előfizetők neve : " + element.getElementsByTagName("Nev").item(*0*).getTextContent());  
 }  
 }  
 } catch (ParserConfigurationException | XPathExpressionException | IOException | SAXException e){  
 e.printStackTrace();  
 }  
 }  
  
}