

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Készítette: **Szemán Péter**

Neptunkód: **ENZ1AT**

A feladat leírása

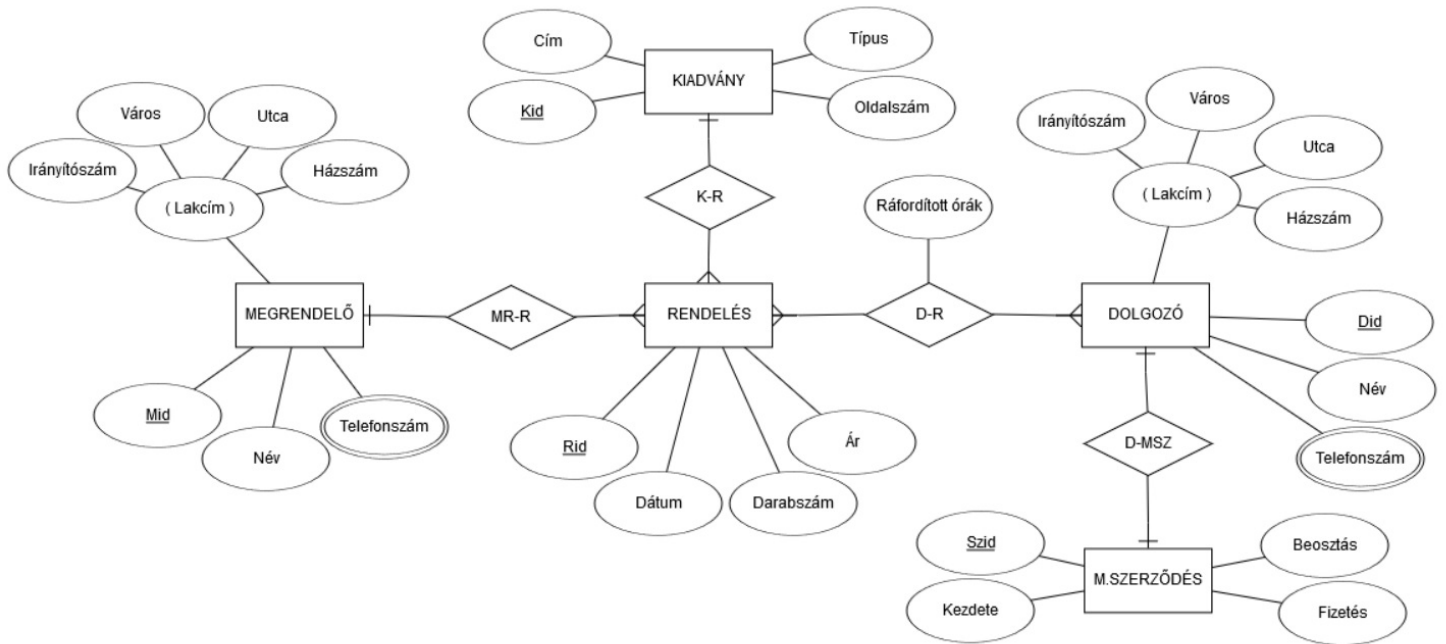
A feladatban egy nyomda működését bemutató XML adatbázist hozok létre. Az adatbázis segítségével tárolhatjuk a nyomdához beérkező rendelések adatait, illetve a kapcsolódó kiadvány, a megrendelő, illetve a munkafolyamatban résztvevő dolgozók adatait is. Az adatbázisban lévő első egyed maga a **kiadvány**. Ez nem más, mint az a könyv, újság, vagy egyéb termék, amelyből a vevő egy bizonyos mennyiséget rendel. A hozzátartozó tulajdonságok közül az azonosító (röviden ID), a cím és az oldalszám magától értetődő. A típus tulajdonságban azt rögzítjük, hogy a kiadvány milyen nyomdai termék, ez lehet többek között újság, könyv, szórólap, hivatalos dokumentum, stb.

A **megrendelő** egyedben és tulajdonságaiban a vevőhöz kapcsolódó adatokat tároljuk. Az azonosító és a név magától értetődő tulajdonságok, míg a lakcím egy összetett tulajdonság, ami négy részből épül fel: irányítószám, város, utca és házszám. Illetve a megrendelőhöz tartozik még egy telefonszám tulajdonság is. Ez többértékű, hogy a vevő több különböző telefonszámát is tárolni tudjuk.

A **dolgozó** egyed hasonló a megrendelőhöz, hiszen itt is egy ember tulajdonságait kell leírni. Emiatt a tulajdonságok is hasonlóak: azonosító, név, lakcím, telefonszám. Ezeken kívül a dolgozóhoz kapcsolódik a **munkaszerződés** nevű egyed is. Ide az egyes dolgozók szerződései, és azoknak adatai kerülnek: azonosító, munkaviszony kezdete, fizetés és beosztás. Mivel minden dolgozóhoz egy szerződés tartozik, és fordítva is, így a két egyed között egy-egy kapcsolat van.

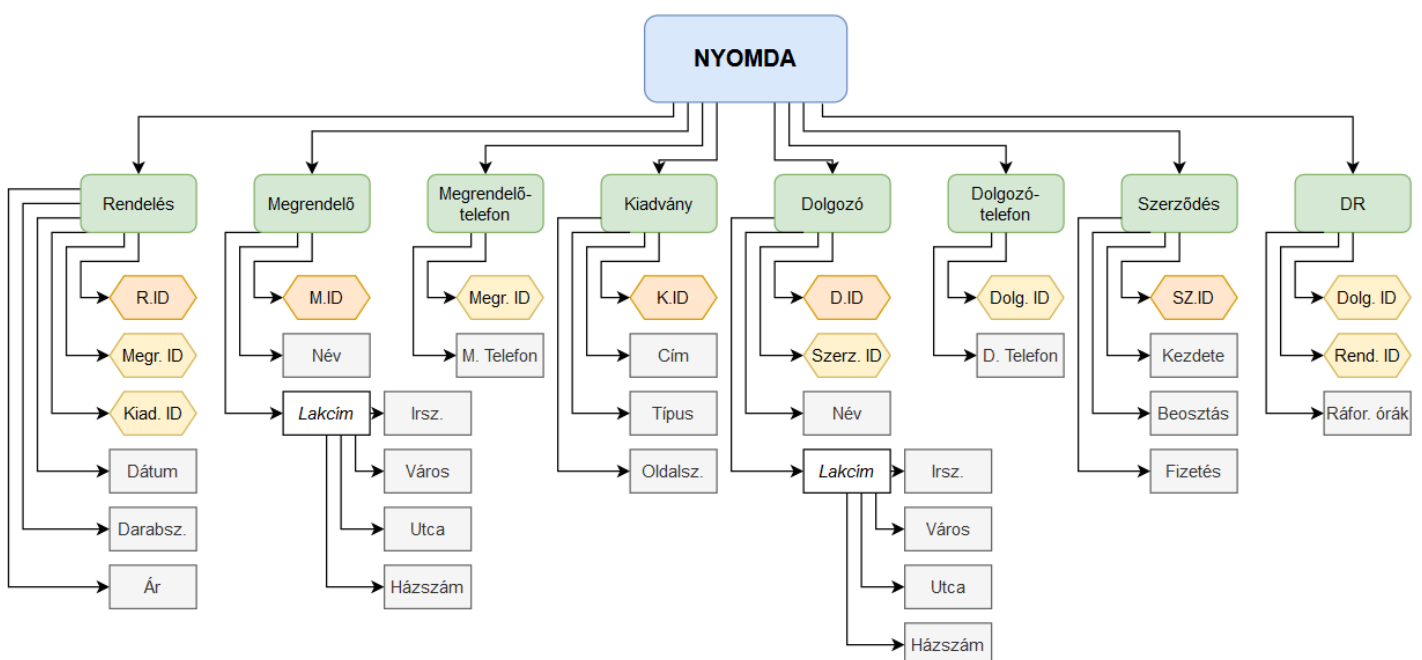
Az adatbázis egyedeit (kiadvány, megrendelő, dolgozó) a **rendelés** nevű egyed köti össze. Nyilvánvaló, hogy minden megrendeléshez tartozik egy megrendelő és egy kiadvány, illetve egy vagy több, a rendelés elkészítésén dolgozó alkalmazott is. A megrendeléshez tartozik még négy másik tulajdonság is. Az azonosító, dátum, darabszám és az ár a nevüknek megfelelő adatokat írják le. A rendelés egyedet a többivel különböző kapcsolatok kötik össze. A kiadvány és a megrendelő egyedeket is egy-több kapcsolat köti össze a rendeléssel. Ez abból következik, hogy egy megrendeléshez csak egy megrendelő és egy kiadvány tartozhat, viszont egy kiadvány több különböző alkalommal is rendelhető, és egy vevő is több különböző megrendelést intézhet a nyomdához. A dolgozót több-több kapcsolat köti a rendeléshez. Egy megrendelés elkészítéséhez több különböző dolgozóra is szükség lehet, egy dolgozó pedig nyilvánvalóan több megrendeléshez is kapcsolódhat. Ennek a kapcsolatnak van egy tulajdonsága is: a ráfordított órák száma. Egy-egy kapcsolat is van a modellben, a korábban már említett dolgozó-munkaszerződés kapcsolat.

Az Adatbázis ER-modellje:



Az ER-modell az *erdplus.com* szerkesztőjével készült. A feladatleírásban tárgyalt egyedeket és tulajdonságokat ábrázoltam az adatbázis rendszerek tantárgyból már tanult módon. A kapcsolatok nevin rövidítettem az általuk összekötött egyedek kezdőbetűinek segítségével.

Az adatbázis konvertálása XDM-modellre:



Az XDM modellt a *draw.io* szerkesztő segítségével készítettem. Röviden az átalakításról: a főbb egyedek, tehát a rendelés, megrendelő, dolgozó, kiadvány, szerződés egyedek itt is külön egységként jelennek meg, tulajdonságaikat megtartják. Az összetett lakcím tulajdonság alkotórészeire bontva jelenik meg. A telefonszám többértékű tulajdonságokból külön egyedek jönnek létre, amik tartalmazzák magát a telefonszámot, illetve a hozzá kapcsolódó tulajdonos azonosítóját idegen kulcsként. A rendelést a megrendelővel és a kiadvánnyal egy-több tulajdonság köti össze, így ezeket egy-egy, a rendelésbe kerülő idegenkulccsal valósítottam meg. A dolgozót a szerződéssel egy-egy kapcsolat köti össze, az idegen kulcsot a dolgozó egyedben helyeztem el. A több-több kapcsolat megvalósításához szintén külön egyed kell, ez lett a DR (dolgozó-rendelés) egyed, ami tartalmazza a két idegen kulcsot és egy további tulajdonságot.

Az ábrán narancssárga hatszöggel jelöltem az elsődleges kulcsokat, citromsárga hatszöggel az idegen kulcsokat, szürke téglalappal pedig az egyéb tulajdonságokat.

Az XML fájl:

```
<?xml version="1.0" encoding="UTF-8"?>

<nyomda
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="XMLSchemaENZ1AT.xsd">

  <rendeles rid="1" kiadvanyid="1" megrendeloid="1">
    <datum>2020-11-14</datum>
    <darab>4</darab>
    <ar>14000</ar>
  </rendeles>

  <rendeles rid="2" kiadvanyid="2" megrendeloid="3">
    <datum>2020-11-29</datum>
    <darab>30</darab>
    <ar>4000</ar>
  </rendeles>

  <rendeles rid="3" kiadvanyid="3" megrendeloid="2">
    <datum>2020-11-30</datum>
    <darab>17</darab>
    <ar>23000</ar>
  </rendeles>

  <megrendelo mid="1">
    <mnev>Kovacs Jakab</mnev>
    <mlakcim>
      <miranyitoszam>3500</miranyitoszam>
      <mvaros>Miskolc</mvaros>
      <utca>Petofi utca</utca>
      <mhazszam>14</mhazszam>
    </mlakcim>
  </megrendelo>

  <megrendelo mid="2">
    <mnev>Paradicsomos Abraham</mnev>
    <mlakcim>
      <miranyitoszam>3910</miranyitoszam>
      <mvaros>Tokaj</mvaros>
```

```

        <mutca>Tiszapart utca</mutca>
        <mhazszam>2</mhazszam>
    </mlakcim>
</megrendelo>

<megrendelo mid="3">
    <mnev>Nagy Ilona</mnev>
    <mlakcim>
        <miranyitoszam>4400</miranyitoszam>
        <mvaros>Nyiregyhaza</mvaros>
        <mutca>Fasor utca</mutca>
        <mhazszam>8</mhazszam>
    </mlakcim>
</megrendelo>

<megrendelotelefon megrendeloid="1">
    <mtelefon>305545546</mtelefon>
</megrendelotelefon>

<megrendelotelefon megrendeloid="2">
    <mtelefon>301155667</mtelefon>
</megrendelotelefon>

<megrendelotelefon megrendeloid="3">
    <mtelefon>201198872</mtelefon>
</megrendelotelefon>

<kiadvany kid="1">
    <cim>Az elveszett almafa</cim>
    <tipus>Konyv</tipus>
    <oldalszam>144</oldalszam>
</kiadvany>

<kiadvany kid="2">
    <cim>Jelentkezési lap</cim>
    <tipus>Dokumentum</tipus>
    <oldalszam>3</oldalszam>
</kiadvany>

<kiadvany kid="3">
    <cim>Tiszaparti hetilap</cim>
    <tipus>Folyoirat</tipus>
    <oldalszam>20</oldalszam>
</kiadvany>

<dolgozo did="1" szerzodesid="1">
    <dnev>Fekete Jolan</dnev>
    <dlakcim>
        <diranyitoszam>3500</diranyitoszam>
        <dvaros>Miskolc</dvaros>
        <dutca>Korte utca</dutca>
        <dhazszam>98</dhazszam>
    </dlakcim>
</dolgozo>

<dolgozo did="2" szerzodesid="2">
    <dnev>Budai David</dnev>
    <dlakcim>
        <diranyitoszam>3500</diranyitoszam>
        <dvaros>Miskolc</dvaros>
        <dutca>Dios utca</dutca>
        <dhazszam>119</dhazszam>
    </dlakcim>
</dolgozo>

<dolgozo did="3" szerzodesid="3">

```

```
<dnev>Sarospataki Elemer</dnev>
<dlakcim>
  <diranyitoszam>3500</diranyitoszam>
  <dvaros>Miskolc</dvaros>
  <utca>Barackfa utca</utca>
  <dhazszam>60</dhazszam>
</dlakcim>
</dolgozo>

<dolgozotelefon dolgozoid="1">
  <dtelefon>305545546</dtelefon>
</dolgozotelefon>

<dolgozotelefon dolgozoid="2">
  <dtelefon>301982345</dtelefon>
</dolgozotelefon>

<dolgozotelefon dolgozoid="3">
  <dtelefon>202066711</dtelefon>
</dolgozotelefon>

<szerezodes szid="1">
  <kezdetek>2020-01-15</kezdetek>
  <beosztas>Gepkezeslo</beosztas>
  <fizetes>250000</fizetes>
</szerezodes>

<szerezodes szid="2">
  <kezdetek>2019-12-01</kezdetek>
  <beosztas>Seged</beosztas>
  <fizetes>210000</fizetes>
</szerezodes>

<szerezodes szid="3">
  <kezdetek>2020-04-07</kezdetek>
  <beosztas>Nyomdai gepmester</beosztas>
  <fizetes>320000</fizetes>
</szerezodes>

<DR dolgozoid="1" rendelesid="1">
  <raforditottorak>5</raforditottorak>
</DR>

<DR dolgozoid="2" rendelesid="2">
  <raforditottorak>3</raforditottorak>
</DR>

<DR dolgozoid="3" rendelesid="3">
  <raforditottorak>4</raforditottorak>
</DR>

</nyomda>
```

Az XMLSchema fájl:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.example.org/XMLSchemaENZ1AZ"
xmlns:tns="http://www.example.org/XMLSchemaENZ1AZ" elementFormDefault="qualified">

  <xs:element name="nyomda">
    <xs:complexType>
      <xs:sequence>

        <xs:element name="rendeles">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="datum" type="xs:date"/>
              <xs:element name="darab" type="xs:integer"/>
              <xs:element name="ar" type="xs:integer"/>
            </xs:sequence>
            <xs:attribute name="rid" type="xs:integer" use="required"/>
            <xs:attribute name="kiadvanyid" type="xs:integer"/>
            <xs:attribute name="megrendeloid" type="xs:integer"/>
          </xs:complexType>
        </xs:element>

        <xs:element name="megrendelo">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="mnev" type="xs:string"/>
              <xs:element name="mlakcim">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="miranyitoszam" type="xs:integer"/>
                    <xs:element name="mvaros" type="xs:string"/>
                    <xs:element name="mutca" type="xs:string"/>
                    <xs:element name="mhazszam" type="xs:integer"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
            <xs:attribute name="mid" type="xs:integer" use="required"/>
          </xs:complexType>
        </xs:element>

        <xs:element name="megrendelotelefon">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="mtelefon" type="xs:integer"/>
            </xs:sequence>
            <xs:attribute name="megrendeloid" type="xs:integer"/>
          </xs:complexType>
        </xs:element>

        <xs:element name="kiadvany">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="cim" type="xs:string"/>
              <xs:element name="tipus" type="xs:string"/>
              <xs:element name="oldalszam" type="xs:integer"/>
            </xs:sequence>
            <xs:attribute name="kid" type="xs:integer" use="required"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</schema>
```

```

</xs:element>

<xs:element name="dolgozo">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="dnev" type="xs:string"/>
      <xs:element name="dlakcim">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="diranyitoszam" type="xs:integer"/>
            <xs:element name="dvaros" type="xs:string"/>
            <xs:element name="dutca" type="xs:string"/>
            <xs:element name="dhazsszam" type="xs:integer"/>

          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="did" type="xs:integer" use="required"/>
    <xs:attribute name="szerzodesid" type="xs:integer"/>
  </xs:complexType>
</xs:element>

<xs:element name="dolgozotelefon">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="dtelefon" type="xs:integer"/>

    </xs:sequence>
    <xs:attribute name="dolgozoid" type="xs:integer"/>
  </xs:complexType>
</xs:element>

<xs:element name="szerzodes">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="kezdetek" type="xs:date"/>
      <xs:element name="beosztas" type="xs:string"/>
      <xs:element name="fizetes" type="xs:integer"/>

    </xs:sequence>
    <xs:attribute name="szid" type="xs:integer" use="required"/>
  </xs:complexType>
</xs:element>

<xs:element name="DR">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="raforditottorak" type="xs:integer"/>

    </xs:sequence>
    <xs:attribute name="dolgozoid" type="xs:integer" use="required"/>
    <xs:attribute name="rendelesid" type="xs:integer" use="required"/>
  </xs:complexType>
</xs:element>

</xs:sequence>
</xs:complexType>

<xs:key name="kid">
  <xs:selector xpath="."/>
  <xs:field xpath="@kid"/>
</xs:key>

<xs:keyref name="kiadvanyidref" refer="kid">
  <xs:selector xpath="."/>

```



```

        <xs:field xpath="@kiadvanyid"/>
    </xs:keyref>

    <xs:key name="mid">
        <xs:selector xpath="."/>
        <xs:field xpath="@mid"/>
    </xs:key>

    <xs:keyref name="megrendeloidref" refer="mid">
        <xs:selector xpath="."/>
        <xs:field xpath="@megrendeloid"/>
    </xs:keyref>

    <xs:keyref name="megrendeloidref2" refer="mid">
        <xs:selector xpath="."/>
        <xs:field xpath="@megrendeloid"/>
    </xs:keyref>

    <xs:key name="rid">
        <xs:selector xpath="."/>
        <xs:field xpath="@rid"/>
    </xs:key>

    <xs:keyref name="rendelesidref" refer="rid">
        <xs:selector xpath="."/>
        <xs:field xpath="@rendelesid"/>
    </xs:keyref>

    <xs:key name="did">
        <xs:selector xpath="."/>
        <xs:field xpath="@did"/>
    </xs:key>

    <xs:keyref name="dolgozoidref" refer="did">
        <xs:selector xpath="."/>
        <xs:field xpath="@dolgozoid"/>
    </xs:keyref>

    <xs:keyref name="dolgozoidref2" refer="did">
        <xs:selector xpath="."/>
        <xs:field xpath="@dolgozoid"/>
    </xs:keyref>

    <xs:key name="szid">
        <xs:selector xpath="."/>
        <xs:field xpath="@szid"/>
    </xs:key>

    <xs:keyref name="szerzodesidref" refer="szid">
        <xs:selector xpath="."/>
        <xs:field xpath="@szerzodesid"/>
    </xs:keyref>

</xs:element>

</schema>

```

A DOM adatbeolvasó program kódja:

```
package enz1atdom2;

import javax.xml.parsers.DocumentBuilderFactory;
import org.w3c.dom.NamedNodeMap;
import javax.xml.parsers.DocumentBuilder;
import org.w3c.dom.Document;
import org.w3c.dom.NodeList;
import org.w3c.dom.Node;
import org.w3c.dom.Element;
import java.io.File;
import java.io.IOException;
import javax.xml.parsers.ParserConfigurationException;
import org.xml.sax.SAXException;

public class DomReadENZ1AT {

    public static void main(String argv[]) {

        try {

            //Létrehozzuk a beolvasáshoz szükséges egységeket
            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
            DocumentBuilder docbuilder = factory.newDocumentBuilder();

            //Beolvassuk az XML fájlt
            Document document = docbuilder.parse(new File("XMLENZ1AT.xml"));
            document.getDocumentElement().normalize();

            //Kiválasztjuk a gyökérelemet
            Element rootElement = document.getDocumentElement();
            System.out.println("Gyokerelem: " + rootElement.getNodeName());

            //Gyerekelemeket eltároljuk egy listában
            NodeList childNodes = rootElement.getChildNodes();

            //Ciklusban végigjárjuk a gyerekelemeket
            for(int i=1; i<childNodes.getLength(); i++) {
                Node node = childNodes.item(i);
                if(node.getNodeType() == Node.ELEMENT_NODE) {
                    System.out.println("-----");
                    System.out.println(node.getNodeName() + ": ");
                    Element element = (Element)node;

                    //Kiíratjuk az attribútumokat (elemjellemzőket)
                    NamedNodeMap attlist = element.getAttributes();

                    for(int j=0; j<attlist.getLength(); j++) {

                        System.out.println(" " +
attlist.item(j).getNodeName() + ": " + attlist.item(j).getNodeValue());
                    }

                    //Kiíratjuk az elemek nevét és értékét
                    Node actualNode = element.getFirstChild();
                    while(actualNode != null) {
                        if(actualNode.getNodeType() ==
Node.ELEMENT_NODE) {
                            Element actualElement =
(Element)actualNode;

                            System.out.println(" " +
actualElement.getNodeName() + ": " + actualElement.getTextContent());
                        }
                        actualNode = actualNode.getNextSibling();
                    }
                }
            }
        }
    }
}
```

```

        }
        actualNode=actualNode.getNextSibling();
    }
}

}

}

//Elvégezzük a kivételkezelést
catch(ParserConfigurationException exception) {
    exception.printStackTrace();
}
catch(IOException exception) {
    exception.printStackTrace();
}
catch(SAXException exception) {
    exception.printStackTrace();
}
}
}
}

```

A DOM adatmódosító program kódja:

```

package enz1atdom2;

import javax.xml.parsers.DocumentBuilderFactory;
import org.w3c.dom.NamedNodeMap;
import javax.xml.parsers.DocumentBuilder;
import org.w3c.dom.Document;
import org.w3c.dom.NodeList;
import org.w3c.dom.Node;
import org.w3c.dom.Element;
import java.io.File;
import java.io.IOException;
import javax.xml.parsers.ParserConfigurationException;

import org.xml.sax.SAXException;

public class DomModifyENZ1AT {

    public static void main(String argv[]) {

        try {

            //Létrehozuk a beolvasáshoz szükséges egységeket
            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
            DocumentBuilder docbuilder = factory.newDocumentBuilder();

            //Beolvassuk az XML fájlt
            Document document = docbuilder.parse(new File("XMLENZ1AT.xml"));
            document.getDocumentElement().normalize();

            //Kiválasztjuk a gyökerelemet
            Element rootElement = document.getDocumentElement();
            System.out.println("Gyokerelem: " + rootElement.getNodeName());

            //Gyerekelemeket eltároljuk egy listában
            NodeList childNodes = rootElement.getChildNodes();

            //Ciklusban végigjárjuk a gyerekelemeket
            for(int i=1; i<childNodes.getLength(); i++) {
                Node node = childNodes.item(i);
                if(node.getNodeType() == Node.ELEMENT_NODE) {

```

```

        System.out.println("-----
");
        System.out.println(node.getNodeName() + ": ");
        Element element = (Element)node;

        //Kiíratjuk az attribútumokat (elemjellemzőket)
        NamedNodeMap attlist = element.getAttributes();

        for(int j=0; j<attlist.getLength(); j++) {

            System.out.println(" " +
attlist.item(j).getNodeName() + ": " + attlist.item(j).getNodeValue());
        }

        //Kiíratjuk az elemek nevét és értékét
        Node actualNode = element.getFirstChild();
        while(actualNode != null) {

            //Kovács Jakab neve rosszul lett regisztrálva,
            //váltassuk meg Kovács Jenőre
            //Kiválasztjuk azt a node-ot, ahol Kovács Jakab
            van
            if(actualNode.getNodeType() == Node.ELEMENT_NODE
&& actualNode.getTextContent().equals("Kovacs Jakab") == true) {

                //Megváltoztatjuk a tartalmat
                actualNode.setTextContent("Kovacs Jeno");
                Element actualElement =
                    (Element)actualNode;

                System.out.println(" " +
actualElement.getNodeName() + ": " + actualElement.getTextContent());
            }
            else if(actualNode.getNodeType() ==
                Node.ELEMENT_NODE) {
                Element actualElement =
                    (Element)actualNode;

                System.out.println(" " +
actualElement.getNodeName() + ": " + actualElement.getTextContent());
            }
            actualNode=actualNode.getNextSibling();
        }
    }

    //Elvégezzük a kivételkezelést
    catch(ParserConfigurationException exception) {
        exception.printStackTrace();
    }
    catch(IOException exception) {
        exception.printStackTrace();
    }
    catch(SAXException exception) {
        exception.printStackTrace();
    }
}
}

```