

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Könyvtár nyilvántartás XML-ben

Készítette: **Kormos Máté**

Neptunkód: **MXM2NQ**

Dátum: **2021. december 09.**

A feladat leírása:

Az adatbázisom egy Könyvtárban dolgozó Dolgozók és az ott tárolt Könyvek, azok Szerzőinek, valamint Kölcsönzőinek adatait tartalmazza. Egyedenként a következő tulajdonságok és kapcsolatok vannak jelen:

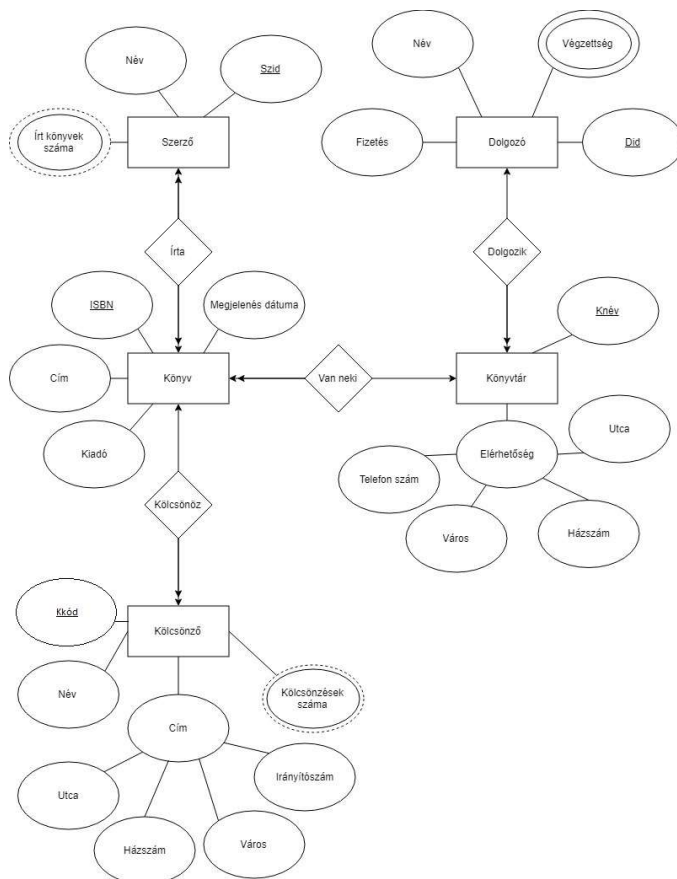
Könyvtár – Dolgozó: 1:N kapcsolat.

Könyv – Könyvtár: 1:N kapcsolat.

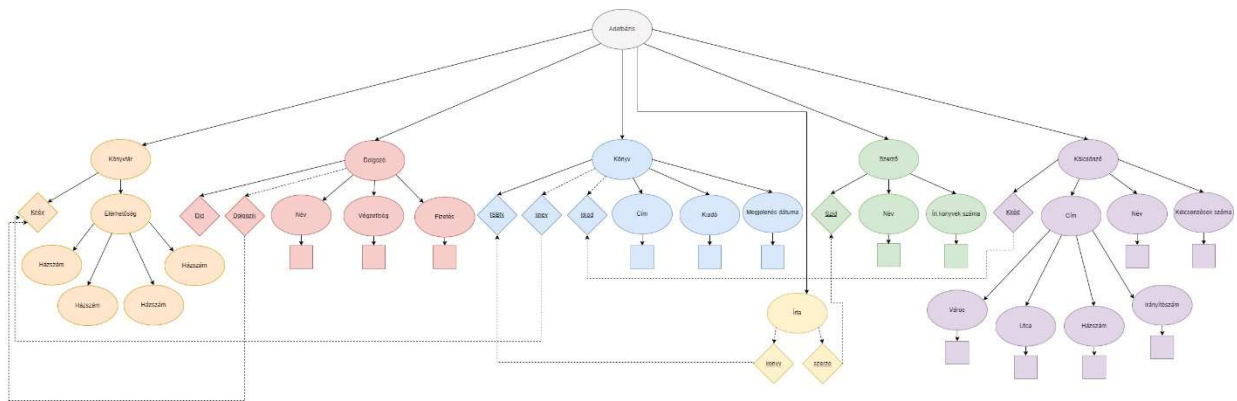
Kölcsönző – Könyv: 1:N kapcsolat.

Szerző – Könyv: N:M kapcsolat.

Minden egyed legalább 3 tulajdonsággal rendelkezik, ebbe beleértve az ID-ket is. Az ER modell alapján elkészítettem az XDM modellt, valamint az XSD és XML dokumentumokat. A javas DOM program írja ki az adatokat és módosítja őket felhasználva az XML fájlt.



XDM modell:



XML Dokumentum:

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

<adatbazis xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="XMLSchemaMXM2NQ.xsd">
<kolcsonzo kkod="1">
    <nev>Horváth Anett</nev>
    <kolcsonzesek_szama>1</kolcsonzesek_szama>
    <cim>
        <irSzam>3300</irSzam>
        <varos>Eger</varos>
        <utca>Szálloda u.</utca>
        <hazszam>6</hazszam>
    </cim>
</kolcsonzo>
<kolcsonzo kkod="2">
    <nev>Erdős László</nev>
    <kolcsonzesek_szama>2</kolcsonzesek_szama>
    <cim>
        <irSzam>3515</irSzam>
        <varos>Miskolc</varos>
        <utca>Mednyánszky László u.</utca>
        <hazszam>11</hazszam>
    </cim>
</kolcsonzo>
<kolcsonzo kkod="3">
    <nev>Hevesi László</nev>
    <kolcsonzesek_szama>0</kolcsonzesek_szama>
    <cim>
        <irSzam>3515</irSzam>
        <varos>Miskolc</varos>
        <utca>Vörösmatry Mihály u.</utca>
        <hazszam>23</hazszam>
    </cim>
```

```
</kolcsonzo>

<konyv ISBN="1234" knev="Bródy Sándor Könyvtár" kkod="1">
  <cim>Harry Potter és a bölcsek köve</cim>
  <kiado>Animus</kiado>
  <megjelenes_datuma>1997</megjelenes_datuma>
</konyv>
<konyv ISBN="1547" knev="Bródy Sándor Könyvtár">
  <cim>Pszichológia</cim>
  <kiado>Osiris</kiado>
  <megjelenes_datuma>2005</megjelenes_datuma>
</konyv>
<konyv ISBN="4167" knev="Miskolci Egyetemi Könyvtár, Levéltár" kkod="2">
  <cim>Valószínűségszámítás</cim>
  <kiado>Nemzedékek Tudása</kiado>
  <megjelenes_datuma>2001</megjelenes_datuma>
</konyv>
<konyv ISBN="5344" knev="Miskolci Egyetemi Könyvtár, Levéltár" kkod="2">
  <cim>Számítógép hálózatok</cim>
  <kiado>Panem Kft.</kiado>
  <megjelenes_datuma>2013</megjelenes_datuma>
</konyv>
<konyv ISBN="4004" knev="Bródy Sándor Könyvtár">
  <cim>Az Aranyember</cim>
  <kiado>Talentum Diákkönyvtár</kiado>
  <megjelenes_datuma>2010</megjelenes_datuma>
</konyv>
<konyv ISBN="7594" knev="Miskolci Egyetemi Könyvtár, Levéltár">
  <cim>Analízis</cim>
  <kiado>Nemzedékek Tudása</kiado>
  <megjelenes_datuma>1988</megjelenes_datuma>
</konyv>

<szerzo szid="1">
  <nev>J.K. Rowling</nev>
  <irt_konyvek_szama>1</irt_konyvek_szama>
</szerzo>
<szerzo szid="2">
  <nev>Richard C. Atkinson</nev>
  <irt_konyvek_szama>1</irt_konyvek_szama>
</szerzo>
<szerzo szid="3">
  <nev>David J. Wetherall</nev>
  <irt_konyvek_szama>1</irt_konyvek_szama>
</szerzo>
<szerzo szid="4">
  <nev>Andrew S. Tanenbaum</nev>
  <irt_konyvek_szama>1</irt_konyvek_szama>
</szerzo>
```

```
<szerzo szid="5">
  <nev>Denkinger Géza</nev>
  <irt_konyvek_szama>2</irt_konyvek_szama>
</szerzo>
<szerzo szid="6">
  <nev>Jókai Mór</nev>
  <irt_konyvek_szama>1</irt_konyvek_szama>
</szerzo>

<konyvtar knev="Bródy Sándor Könyvtár">
  <elerhetoseg>
    <telefon_szam>(36) 516-632</telefon_szam>
    <varos>Eger</varos>
    <utca>Kossuth u.</utca>
    <hazszam>16</hazszam>
  </elerhetoseg>
</konyvtar>
<konyvtar knev="Miskolci Egyetemi Könyvtár, Levéltár">
  <elerhetoseg>
    <telefon_szam>+36-46-565-320</telefon_szam>
    <varos>Miskolc</varos>
    <utca>Egyetemváros</utca>
    <hazszam>1</hazszam>
  </elerhetoseg>
</konyvtar>

<dolgozo did="1" knev="Bródy Sándor Könyvtár">
  <nev>Kis Anna</nev>
  <fizetes>100000</fizetes>
  <vegzettseg>8 általános</vegzettseg>
  <vegzettseg>Érettségi</vegzettseg>
</dolgozo>
<dolgozo did="2" knev="Bródy Sándor Könyvtár">
  <nev>Halász Péter</nev>
  <fizetes>120000</fizetes>
  <vegzettseg>Érettségi</vegzettseg>
  <vegzettseg>Informatikus OKJ</vegzettseg>
</dolgozo>
<dolgozo did="3" knev="Miskolci Egyetemi Könyvtár, Levéltár">
  <nev>Kovács Zsófia</nev>
  <fizetes>150000</fizetes>
  <vegzettseg>8 általános</vegzettseg>
  <vegzettseg>Érettségi</vegzettseg>
  <vegzettseg>Egyetemi Bsc Diploma</vegzettseg>
</dolgozo>
<dolgozo did="4" knev="Miskolci Egyetemi Könyvtár, Levéltár">
  <nev>Beregi Júlia</nev>
  <fizetes>110000</fizetes>
  <vegzettseg>8 általános</vegzettseg>
```

```

    <vegzettseg>Érettségi</vegzettseg>
</dolgozo>

<irta szerzo="1" konyv="1234" />
<irta szerzo="2" konyv="1547" />
<irta szerzo="5" konyv="4167" />
<irta szerzo="3" konyv="5344" />
<irta szerzo="4" konyv="5344" />
<irta szerzo="6" konyv="4004" />
<irta szerzo="5" konyv="7594" />

</adatbazis>

```

XML Schema:

```

<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="adatbazis">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="kolcsonzo" maxOccurs="unbounded" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="nev"> </xs:element>
              <xs:element type="xs:byte" name="kolcsonzesek_szama">
</xs:element>
                <xs:element name="cim">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element type="iranyitoszam" name="irSzam">
</xs:element>
                        <xs:element type="xs:string" name="varos"> </xs:element>
                        <xs:element type="xs:string" name="utca"> </xs:element>
                        <xs:element type="xs:string" name="hazszam"> </xs:element>
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
                <xs:attribute type="xs:byte" name="kkod" use="required">
</xs:attribute>
              </xs:complexType>
            </xs:element>
          <xs:element name="konyv" maxOccurs="unbounded" minOccurs="1">
            <xs:complexType>
              <xs:sequence>
                <xs:element type="xs:string" name="cim"> </xs:element>
                <xs:element type="xs:string" name="kiado"> </xs:element>

```

```

        <xs:element type="xs:short" name="megjelenes_datuma">
</xs:element>
        </xs:sequence>
        <xs:attribute type="xs:short" name="ISBN" use="required">
</xs:attribute>
        <xs:attribute type="xs:string" name="knev" use="required">
</xs:attribute>
        <xs:attribute type="xs:short" name="kkod" use="optional">
</xs:attribute>
        </xs:complexType>
</xs:element>
<xs:element name="szerzo" maxOccurs="unbounded" minOccurs="1">
    <xs:complexType>
        <xs:sequence>
            <xs:element type="xs:string" name="nev"> </xs:element>
            <xs:element type="xs:short" name="irt_konyvek_szama">
</xs:element>
        </xs:sequence>
        <xs:attribute type="xs:short" name="szid" use="required">
</xs:attribute>
    </xs:complexType>
</xs:element>
<xs:element name="konyvtar" maxOccurs="unbounded" minOccurs="1">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="elerhetoseg">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element type="xs:string" name="telefon_szam">
</xs:element>
                        <xs:element type="xs:string" name="varos">
</xs:element>
                        <xs:element type="xs:string" name="utca">
</xs:element>
                        <xs:element type="xs:string" name="hazzsam">
</xs:element>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
        <xs:attribute type="xs:string" name="knev" use="required">
</xs:attribute>
    </xs:complexType>
</xs:element>
<xs:element name="dolgozo" maxOccurs="unbounded" minOccurs="1">
    <xs:complexType>
        <xs:sequence>
            <xs:element type="xs:string" name="nev"> </xs:element>
            <xs:element type="xs:int" name="fizetes"> </xs:element>

```

```

        <xs:element type="vegzettseg" name="vegzettseg"
maxOccurs="unbounded"> </xs:element>
    </xs:sequence>
    <xs:attribute type="xs:short" name="did" use="required">
</xs:attribute>
    <xs:attribute type="xs:string" name="knev" use="required">
</xs:attribute>
    </xs:complexType>
</xs:element>
    <xs:element name="irta" maxOccurs="unbounded">
    <xs:complexType>
        <xs:attribute type="xs:short" name="szerzo" use="required">
</xs:attribute>
        <xs:attribute type="xs:short" name="konyv" use="required">
</xs:attribute>
    </xs:complexType>
    </xs:element>
</xs:sequence>
</xs:complexType>
<xs:key name="kkod">
    <xs:selector xpath="kolcsonzo"></xs:selector>
    <xs:field xpath="@kkod"></xs:field>
</xs:key>
<xs:key name="ISBN">
    <xs:selector xpath="konyv"></xs:selector>
    <xs:field xpath="@ISBN"></xs:field>
</xs:key>
<xs:key name="knev">
    <xs:selector xpath="konyvtar"></xs:selector>
    <xs:field xpath="@knev"></xs:field>
</xs:key>
<xs:key name="did">
    <xs:selector xpath="dolgozo"></xs:selector>
    <xs:field xpath="@did"></xs:field>
</xs:key>
<xs:key name="szid">
    <xs:selector xpath="szerzo"></xs:selector>
    <xs:field xpath="@szid"></xs:field>
</xs:key>

<xs:keyref name="fk_konyvtar_dolgozo" refer="knev">
    <xs:selector xpath="dolgozo"></xs:selector>
    <xs:field xpath="@konyvtar"></xs:field>
</xs:keyref>
<xs:keyref name="fk_konyv_kolcsonzo" refer="kkod">
    <xs:selector xpath="konyv"></xs:selector>
    <xs:field xpath="@kkod"></xs:field>
</xs:keyref>
<xs:keyref name="fk_konyv_konyvtar" refer="knev">

```



```

        <xs:selector xpath="konyv"></xs:selector>
        <xs:field xpath="@knev"></xs:field>
    </xs:keyref>
    <xs:keyref name="fk_dolgozo_konyvtar" refer="knev">
        <xs:selector xpath="dolgozo"></xs:selector>
        <xs:field xpath="@knev"></xs:field>
    </xs:keyref>
    <xs:keyref name="fk_irta_szerzo" refer="szid">
        <xs:selector xpath="irta"></xs:selector>
        <xs:field xpath="@szerzo"></xs:field>
    </xs:keyref>
    <xs:keyref name="fk_irta_konyv" refer="ISBN">
        <xs:selector xpath="irta"></xs:selector>
        <xs:field xpath="@konyv"></xs:field>
    </xs:keyref>
</xs:element>
<xs:simpleType name="iranyitoszam">
    <xs:restriction base="xs:string">
        <xs:pattern value="[1-9]{1}[0-9]{3}"></xs:pattern>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="vegzettseg">
    <xs:restriction base="xs:string">
        <xs:pattern value="Nyelvvizsga"></xs:pattern>
        <xs:pattern value="8 általános"></xs:pattern>
        <xs:pattern value="Érettségi"></xs:pattern>
        <xs:pattern value="Egyetemi Bsc Diploma"></xs:pattern>
        <xs:pattern value="Informatikus OKJ"></xs:pattern>
    </xs:restriction>
</xs:simpleType>
</xs:schema>

```

Adatolvasás:

Ez a program kiolvassa az XMLMXM2NQ.xml fájlból. Kiestázza az adatbázisban található könyveket és kölcsönzőket.

```

import java.io.File;
import java.io.IOException;

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 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 DOMReadMXM2NQ {
    public static void main(String arg[])
        throws SAXException, IOException, ParserConfigurationException,
XPathExpressionException {
        // input file megnyitása
        File xmlFile = new File("../XMLMXM2NQ.xml");

        // dbuilder létrehozása
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = factory.newDocumentBuilder();

        // dokumentum beolvasása
        Document doc = dBuilder.parse(xmlFile);
        doc.getDocumentElement().normalize();

        // gyökér elem
        System.out.println("Gyökér elem: " +
doc.getDocumentElement().getNodeName());
        System.out.println("\n");

        XPath xPath = XPathFactory.newInstance().newXPath();

        // könyv beolvasása
        NodeList nList = doc.getElementsByTagName("konyv");

        for (int i = 0; i < nList.getLength(); i++) {
            Node nNode = nList.item(i);

            if (nNode.getNodeType() != Node.ELEMENT_NODE)
                continue;

            Element eElement = (Element) nNode;
            String ISBN = eElement.getAttribute("ISBN");
            // Könyv adatai
            System.out.println("konyv Id: " + ISBN);

            System.out.println("Könyv címe: " +
                eElement.getElementsByTagName("cim").item(0).getTextContent());
            System.out.println("Könyv kiadója: " +
                eElement.getElementsByTagName("kiado").item(0).getTextContent());
            System.out.println("Könyv megjelenési dátuma: " +
                eElement.getElementsByTagName("megjelenes_datuma").item(0).getTextCo
ntent());

```

```

        String szerzo_expression = "//szerzo[@szid=//irta[@konyv=" + ISBN +
        "]/@szerzo]/nev";
        NodeList szerzo_nevek = (NodeList)
        XPath.compile(szerzo_expression).evaluate(doc, XPathConstants.NODESET);

        for (int j = 0; j < szerzo_nevek.getLength(); j++) {
            Node n = szerzo_nevek.item(j);

            if (n.getNodeType() != Node.ELEMENT_NODE)
                continue;

            System.out.println("szerzo: " + n.getTextContent());
        }

        System.out.println();
    }
    System.out.println("_____");

    NodeList nList2 = doc.getElementsByTagName("kolcsonzo");
    for (int i = 0; i < nList2.getLength(); i++) {
        Node nNode = nList2.item(i);

        if (nNode.getNodeType() != Node.ELEMENT_NODE)
            continue;

        Element eElement = (Element) nNode;
        // Kölcsönző adatai
        System.out.println("kolcs Id: " + eElement.getAttribute("kkod"));
        System.out.println("Kölcsönző neve: " +
            eElement.getElementsByTagName("nev").item(0).getTextContent());
        System.out.println("Kölcsönző könyveinek száma: " +
            eElement.getElementsByTagName("kolcsonzesek_szama").item(0).getTextC
            ontent());
        System.out.println("Kölcsönző címe: " +
            eElement.getElementsByTagName("cim").item(0).getTextContent());

        System.out.println();
    }

    System.out.println("_____");
}
}

```

Adatlekérdezés:

Ez a program lekérdezi az XMLMXM2NQ.xml fájlból. Lekérdezi a miskolci kölcsönzők neveit és a Denkinger Géza által írt könyvek címeit.

```
import java.io.File;
import java.io.IOException;

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 org.w3c.dom.Document;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;

public class DOMQueryMXM2NQ {

    public static void main(String arg[])
        throws SAXException, IOException, ParserConfigurationException,
        XPathExpressionException {

        //input file megnyitása
        File inputFile = new File("../XMLMXM2NQ.xml");

        //dbuilder létrehozása
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = factory.newDocumentBuilder();

        //dokumentum beolvasása
        Document doc = dBuilder.parse(inputFile);
        doc.getDocumentElement().normalize();

        System.out.println("Miskolci kölcsönzők nevei");
        listKolcsonzokByCity("Miskolc", doc);
        System.out.println("");

        System.out.println("Denkinger Géza könyveinek címei");
        listKonyvCimekBySzerzo("Denkinger Géza", doc);
        System.out.println("");

    }
}
```

```

    private static void listKolcsonzokByCity(String city, Document doc) throws
XPathExpressionException {
        // XPath fa létrehozása
        XPath xPath = XPathFactory.newInstance().newXPath();

        String expression = "//kolcsonzo[./cim/varos='" + city + "']/nev";
        NodeList nodeList = (NodeList) xPath.compile(expression).evaluate(doc,
XPathConstants.NODESET);

        for (int j = 0; j < nodeList.getLength(); j++) {
            Node n = nodeList.item(j);

            if (n.getNodeType() != Node.ELEMENT_NODE)
                continue;

            System.out.println("Kölcsönző neve: " + n.getTextContent());
        }
    }

    private static void listKonyvCimekBySzerzo(String szerzoNev, Document doc)
throws XPathExpressionException {
        // XPath fa létrehozása
        XPath xPath = XPathFactory.newInstance().newXPath();

        String expression = "//konyv[@ISBN=//irta[@szerzo=//szerzo[./nev='" +
szerzoNev + "']/@szid]/@konyv]/cim";

        NodeList nodeList = (NodeList) xPath.compile(expression).evaluate(doc,
XPathConstants.NODESET);
        for (int j = 0; j < nodeList.getLength(); j++) {
            Node n = nodeList.item(j);

            if (n.getNodeType() != Node.ELEMENT_NODE)
                continue;

            System.out.println("Könyv címe: " + n.getTextContent());
        }
    }
}

```

Adatmódosítás:

Ez a program módosítja az XMLMXM2NQ.xml fájl tartalmát és annak eredményét kiírja az XMLMXM2NQ.out.xml-be. A „Kis Anna” nevű könyvtári dolgozóhoz hozzáad egy új, „Nyelvvizsga” nevű végzettséget és a „Hevesi László” nevű kölcsönző címében a házszámot „23”-ról „666”-ra módosítja.

```
import java.io.File;
import java.io.IOException;
import java.io.UnsupportedEncodingException;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.transform.OutputKeys;
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 javax.xml.xpath.XPath;
import javax.xml.xpath.XPathConstants;
import javax.xml.xpath.XPathExpressionException;
import javax.xml.xpath.XPathFactory;

import org.w3c.dom.Document;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;

public class DOMModifyMXM2NQ {
    public static void main(String arg[]) throws IOException,
        ParserConfigurationException, SAXException,
        TransformerConfigurationException, TransformerException,
        XPathExpressionException {
        // Input output file-ok
        File inputXMLFile = new File("../XMLMXM2NQ.xml");
        File outputXMLFile = new File("../XMLMXM2NQ.out.xml");
        // dbuilder létrehozása
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = factory.newDocumentBuilder();

        // dokumentum beolvasása
        Document inputDoc = dBuilder.parse(inputXMLFile);
        inputDoc.getDocumentElement().normalize();

        // Módosítások végrehajtása
        // Végzettség hozzáadása az egyik dolgozóhoz
```

```

        addDolgozoVegzettseg("1", "Nyelvvizsga", inputDoc);
        // Kölcsönző házzámának a módosítása
        changeKolcsonzoHazzsam("3", "666", inputDoc);

        // Kiíratás console-ba és file-ba
        write(inputDoc, outputXMLFile);
    }

    private static void write(Document doc, File outputFile)
        throws TransformerException, UnsupportedEncodingException {
        TransformerFactory transformerFactory =
TransformerFactory.newInstance();
        Transformer transf = transformerFactory.newTransformer();
        transf.setOutputProperty(OutputKeys.ENCODING, "UTF-8");
        transf.setOutputProperty(OutputKeys.INDENT, "yes");
        transf.setOutputProperty("{http://xml.apache.org/xslt}indent-amunt",
"2");

        DOMSource source = new DOMSource(doc);

        StreamResult console = new StreamResult(System.out);
        StreamResult file = new StreamResult(outputFile);

        transf.transform(source, console);
        transf.transform(source, file);
    }

    private static void addDolgozoVegzettseg(String dolgozoId, String
vegzettseg, Document doc)
        throws XPathExpressionException {
        XPath xPath = XPathFactory.newInstance().newXPath();
        String expression = "//dolgozo[@did='" + dolgozoId + "']";
        NodeList nodeList = (NodeList) xPath.compile(expression).evaluate(doc,
XPathConstants.NODESET);

        for (int i = 0; i < nodeList.getLength(); i++) {
            Node n = nodeList.item(i);

            Node vegzettsegNode = doc.createElement("vegzettseg");
            vegzettsegNode.setTextContent(vegzettseg);
            n.appendChild(vegzettsegNode);
        }
    }

    private static void changeKolcsonzoHazzsam(String kolcsonzoId, String
hazzsam, Document doc)
        throws XPathExpressionException {
        XPath xPath = XPathFactory.newInstance().newXPath();

```

```
String expression = "//kolcsonzo[@kkod='" + kolcsonzoId +
"']/cim/hazszam";
NodeList nodeList = (NodeList) xpath.compile(expression).evaluate(doc,
XPathConstants.NODESET);

for (int i = 0; i < nodeList.getLength(); i++) {
    Node n = nodeList.item(i);

    n.setTextContent(hazszam);
}
}
```