

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Készítette: **Nagy Dávid**

Neptunkód: **BQCMAU**

Dátum: **2020. december 01.**

A feladat leírása:

1. Feladat

a) Leírás:

Az adatbázisom egy Telepen dolgozó Gazdák és az ott kezelt állatok adatainkat kezelésére szolgál.

Egyedenként megtalálhatóak a következő tulajdonságok és kapcsolatok:

Gazda – Telep: 1:1 kapcsolat.

Telep – KezeltÁllat: 1:N kapcsolat.

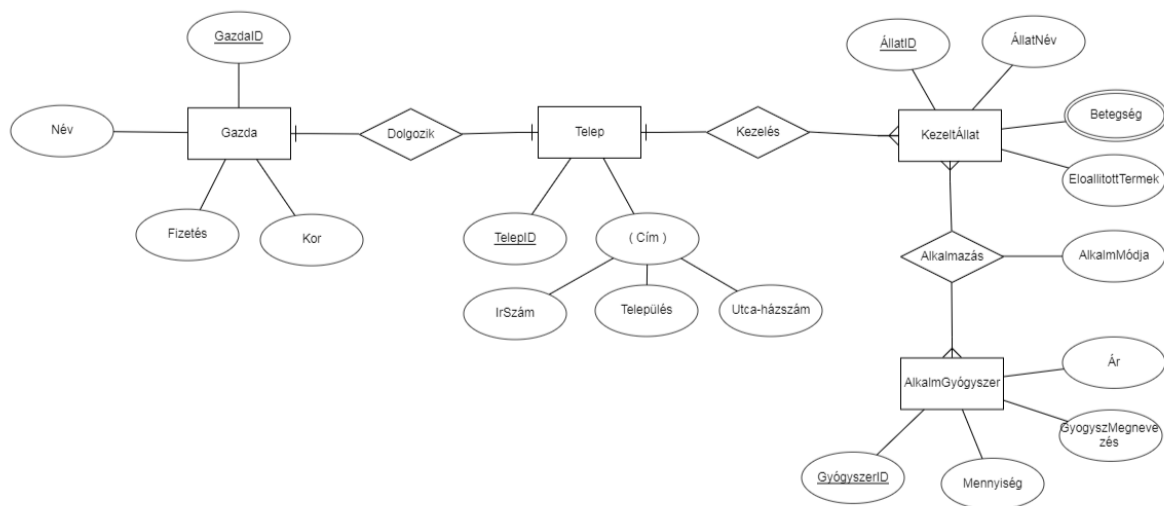
KezeltÁllat – AlkalmGyógyszer: N:M kapcsolat, ahol a kapcsolat is kapott 1 tulajdonságot.

Minden egyed legalább 4 tulajdonsággal van ellátva, ebbe beleértve az ID-ket is.

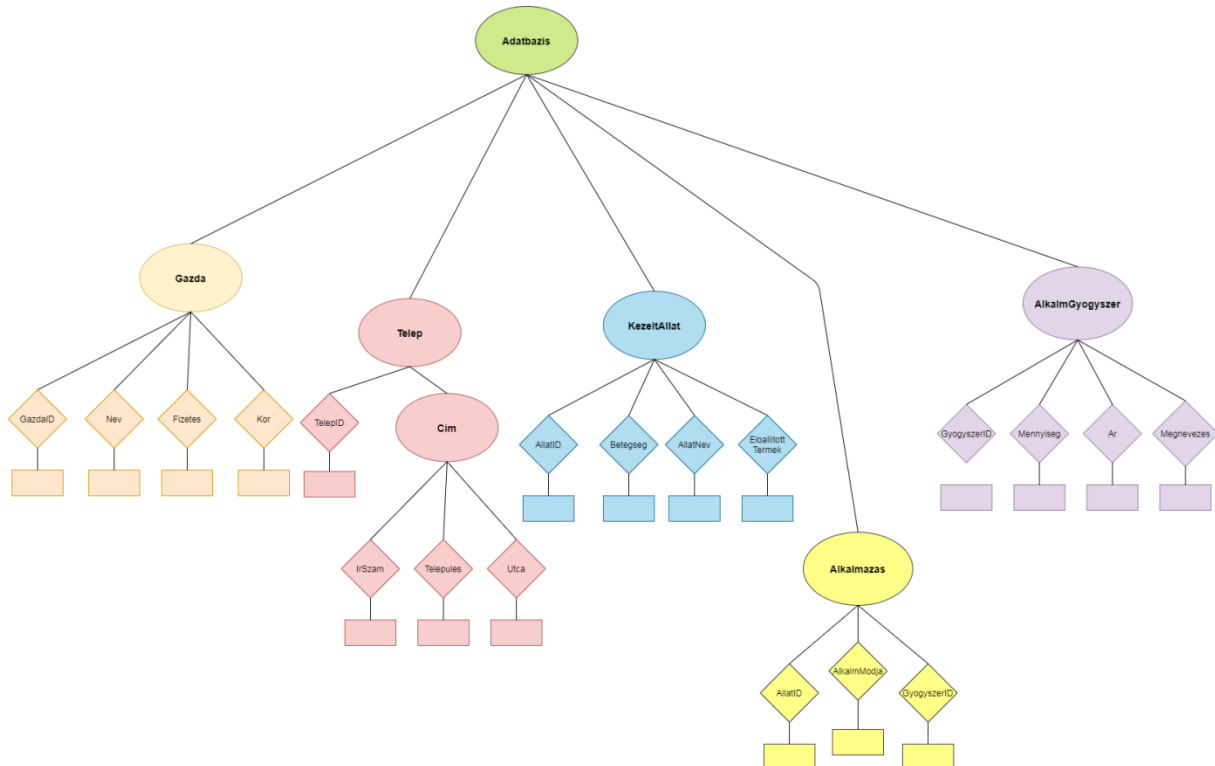
Természetesen az elkészült ER modell alapján elkészült az XDM, az XML és az XSD dokumentum.

A java programok szolgálnak az adatok kiírására, és módosítására felhasználva az XML fájlt.

ER modell:



b) XDM modell:



c) XML dokumentum:

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

```
<adatbazis
```

```
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
  xsi:schemaLocation="C:\Users\Pirella\eclipse-
workspace\DOMParseBQCMAU\src\hu\domparse\bqcmdu\XMLSchemaBQCMAU.xsd">
```

```
  <telep id="01">
```

```
    <cim>
```

```
      <irSzam>3535</irSzam>
```

```
      <telepules>Budapest</telepules>
```

```
      <utca>Kutya</utca>
```

```
    </cim>
```

```
  </telep>
```

```
  <gazda id="001">
```

```
    <nev>Péter</nev>
```

<fizetes>1000</fizetes>

<kor>32</kor>

</gazda>

<gazda id="002">

<nev>István</nev>

<fizetes>2000</fizetes>

<kor>35</kor>

</gazda>

<kezeltAllat id="1000">

<betegseg>Megfázás</betegseg>

<allatNev>Sertés</allatNev>

<eloallitottTermek>Hús</eloallitottTermek>

</kezeltAllat>

<kezeltAllat id="1001">

<betegseg>TBC</betegseg>

<allatNev>Marha</allatNev>

<eloallitottTermek>Hús</eloallitottTermek>

</kezeltAllat>

<alkalmGyogyszer id="901">

<mennyiseg>1</mennyiseg>

<ar>12000</ar>

<megnevezes>Origo</megnevezes>

</alkalmGyogyszer>

```

<alkalmGyogyszer id="902">
    <mennyiseg>2</mennyiseg>
    <ar>35000</ar>
    <megnevezes>Sentor</megnevezes>
</alkalmGyogyszer>

```

```

<alkalmazas kezeltAllatId="1000" gyógyszerId="901">
    <alkalmazasModja>injekció</alkalmazasModja>
</alkalmazas>

```

```

<alkalmazas kezeltAllatId="1001" gyógyszerId="902">
    <alkalmazasModja>tabletta</alkalmazasModja>
</alkalmazas>

```

```

</adatbazis>

```

d) XMLSchema dokumentum:

```

<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="telep">

```

```

<xs:complexType>

```

```

<xs:sequence>

```

```

<xs:element name="cim">

```

```

<xs:complexType>

```

```

<xs:sequence>

```

```

<xs:element type="xs:short" name="irSzam"> </xs:element>

```

```
<xs:element type="xs:string" name="telepules"> </xs:element>

<xs:element type="xs:string" name="utca"> </xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:sequence>

<xs:attribute type="xs:byte" name="id"> </xs:attribute>

</xs:complexType>

</xs:element>

<xs:element name="gazda" maxOccurs="unbounded" minOccurs="0">

  <xs:complexType>

    <xs:sequence>

      <xs:element type="xs:string" name="nev"> </xs:element>

      <xs:element type="xs:short" name="fizetes"> </xs:element>

      <xs:element type="xs:byte" name="kor"> </xs:element>

    </xs:sequence>

    <xs:attribute type="xs:byte" name="id" use="optional"> </xs:attribute>

  </xs:complexType>

</xs:element>

<xs:element name="kezeltAllat" maxOccurs="unbounded" minOccurs="0">

  <xs:complexType>

    <xs:sequence>

      <xs:element type="xs:string" name="betegseg"> </xs:element>

      <xs:element type="xs:string" name="allatNev"> </xs:element>

      <xs:element type="xs:string" name="eloallitottTermek"> </xs:element>

    </xs:sequence>

    <xs:attribute type="xs:short" name="id" use="optional"> </xs:attribute>
```

```

</xs:complexType>

</xs:element>

<xs:element name="alkalmGyogyszer" maxOccurs="unbounded" minOccurs="0">

  <xs:complexType>

    <xs:sequence>

      <xs:element type="xs:byte" name="mennyiseg"> </xs:element>

      <xs:element type="xs:int" name="ar"> </xs:element>

      <xs:element type="xs:string" name="megnevezes"> </xs:element>

    </xs:sequence>

    <xs:attribute type="xs:short" name="id" use="optional"> </xs:attribute>

  </xs:complexType>

</xs:element>

<xs:element name="alkalmazas" maxOccurs="unbounded" minOccurs="0">

  <xs:complexType>

    <xs:sequence>

      <xs:element type="xs:string" name="alkalmazasModja"> </xs:element>

    </xs:sequence>

    <xs:attribute type="xs:short" name="kezeltAllatId" use="optional"> </xs:attribute>

    <xs:attribute type="xs:short" name="gyogyszerId" use="optional"> </xs:attribute>

  </xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:schema>

```

2. Feladat

A feladat egy DOM program készítése az XML dokumentum adatainak adminisztrálása alapján:

a) adatolvasás

```
package hu.domparse.bqcmau;

import java.io.*;
import javax.xml.parsers.*;
import javax.xml.xpath.*;
import org.w3c.dom.*;
import org.xml.sax.*;

public class DOMReadBQCMAU {

    public static void main(String[] args)

        throws SAXException, IOException, ParserConfigurationException,
XPathExpressionException {

        try {

            File inputFile = new File(

                " XMLBQCMAU.xml");

            DocumentBuilderFactory dbFactory =
DocumentBuilderFactory.newInstance();

            DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();

            Document doc = dBuilder.parse(inputFile);

            doc.getDocumentElement().normalize();

            // Gyökérelem név kiírása

            System.out.println("Egyed neve: \t" +
doc.getDocumentElement().getNodeName());

            NodeList nList = doc.getElementsByTagName("telep");

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

                Node nNode = nList.item(i);

                // Telep egyed név kiírása

                System.out.println("\nEgyed neve: \t" + nNode.getNodeName());

                if (nNode.getNodeType() == Node.ELEMENT_NODE) {
```



```

        Element eElement = (Element) nNode;

        // Telep egyes elemeinek kiírása
        System.out.println("telep id: " + eElement.getAttribute("id"));
    }
}

nList = doc.getElementsByTagName("cim");
for (int i = 0; i < nList.getLength(); i++) {
    Node nNode = nList.item(i);
    System.out.println("\nElem neve: \t" + nNode.getNodeName());
    if (nNode.getNodeType() == Node.ELEMENT_NODE) {
        Element eElement = (Element) nNode;
        // cim elem elemeinek kiírása
        System.out.println("irSzam: \t" +
eElement.getElementsByTagName("irSzam").item(i).getTextContent());
        System.out.println(
                                "telepules: \t" +
eElement.getElementsByTagName("telepules").item(i).getTextContent());
        System.out.println("utca: \t\t" +
eElement.getElementsByTagName("utca").item(i).getTextContent());
    }
}

nList = doc.getElementsByTagName("gazda");
for (int i = 0; i < nList.getLength(); i++) {
    Node nNode = nList.item(i);
    // Gazda egyes név kiírása
    System.out.println("\nEgyed neve: \t" + nNode.getNodeName());
    if (nNode.getNodeType() == Node.ELEMENT_NODE) {
        Element eElement = (Element) nNode;
        // gazda egyes elemeinek kiírása

```

```

        System.out.println("gazda id: \t" +
eElement.getAttribute("id"));

        System.out.println("nev: \t\t" +
eElement.getElementsByTagName("nev").item(0).getTextContent());

        System.out
            .println("fizetes: \t" +
eElement.getElementsByTagName("fizetes").item(0).getTextContent());

        System.out.println("kor: \t\t" +
eElement.getElementsByTagName("kor").item(0).getTextContent());

    }

}

nList = doc.getElementsByTagName("kezeltAllat");
for (int i = 0; i < nList.getLength(); i++) {

    Node nNode = nList.item(i);

    // KezeltAllat egyed név kiírása

    System.out.println("\nEgyed neve: \t" + nNode.getNodeName());

    if (nNode.getNodeType() == Node.ELEMENT_NODE) {

        Element eElement = (Element) nNode;

        // kezeltAllat egyed elemeinek kiírása

        System.out.println("kezeltAllatId: \t" +
eElement.getAttribute("id"));

        System.out.println(

            "betegseg: \t\t" +
eElement.getElementsByTagName("betegseg").item(0).getTextContent());

        System.out.println(

            "allatNev: \t\t" +
eElement.getElementsByTagName("allatNev").item(0).getTextContent());

        System.out.println("eloallitottTermek: \t"

            +

eElement.getElementsByTagName("eloallitottTermek").item(0).getTextContent());

    }

}

```

```

nList = doc.getElementsByTagName("alkalmGyogyszer");

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

    Node nNode = nList.item(i);

    // alkalmGyogyszer egyed név kiírása
    System.out.println("\nEgyed neve: \t" + nNode.getNodeName());

    if (nNode.getNodeType() == Node.ELEMENT_NODE) {

        Element eElement = (Element) nNode;

        // alkalmGyogyszer egyed elemeinek kiírása
        System.out.println("gyogyszerId: \t" +
eElement.getAttribute("id"));

        System.out.println(

                                "mennyiseg: \t" +
eElement.getElementsByTagName("mennyiseg").item(0).getTextContent());

        System.out.println("ar: \t\t" +
eElement.getElementsByTagName("ar").item(0).getTextContent());

        System.out.println(

                                "megnevezes: \t" +
eElement.getElementsByTagName("megnevezes").item(0).getTextContent());

    }

}

```

```

nList = doc.getElementsByTagName("alkalmazas");

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

    Node nNode = nList.item(i);

    // alkalmazas kapcsolat név kiírása
    System.out.println("\nEgyed neve: \t" + nNode.getNodeName());

    if (nNode.getNodeType() == Node.ELEMENT_NODE) {

        Element eElement = (Element) nNode;

        // alkalmGyogyszer kapcsolat elemeinek kiírása
        System.out.println("kezeltAllatId: \t\t" +
eElement.getAttribute("kezeltAllatId")

```

```

                                + "\ngyogyszerId \t\t" +
eElement.getAttribute("gyogyszerId"));

                                System.out.println("alkalmazasModja: \t"
                                +
eElement.getElementsByTagName("alkalmazasModja").item(0).getTextContent());

                                }

                                }

                                } catch (Exception e) {

                                e.printStackTrace();

                                }

                                }

}

```

b) adatmódosítás

```

package hu.domp.parse.bqcm.au;

import java.io.*;
import java.text.ParseException;
import java.time.*;
import java.time.format.*;

import javax.xml.parsers.*;
import javax.xml.xpath.*;

import org.w3c.dom.*;
import org.w3c.dom.traversal.*;
import org.xml.sax.*;

public class DOMModifyBQCMAU {

    public static void main(String[] args) throws ParserConfigurationException, SAXException,
    IOException,

        XPathExpressionException, DOMException, ParseException {

        // XML fájl beolvasása

```

```

File xml = new File("XMLBQCMAU.xml");

// XML fájl DOM document való formában való alakítása
DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
DocumentBuilder builder = factory.newDocumentBuilder();
Document document = builder.parse(xml);

DomModifier.modifyDom(document);

DocumentTraversal traversal = (DocumentTraversal) document;

// DOM TreeWalker inicializálása

TreeWalker walker = traversal.createTreeWalker(document.getDocumentElement(),
        NodeFilter.SHOW_ELEMENT | NodeFilter.SHOW_TEXT, null, true);

// a DOM bejárása
DomTraverser.traverseLevel(walker, "");
}

private static class DomModifier {

    public static void modifyDom(Document document) throws XPathExpressionException,
    DOMException, ParseException {

        XPathFactory factory = XPathFactory.newInstance();

        XPath xpath = factory.newXPath();

        //Az alkalmazasnál ahol 1000 a kezeltAllatId, ott módosítsa a gyógyszerId-t '913'-ra
        Node csapat = (Node) xpath.evaluate("//alkalmazas[./@kezeltAllatId='1000']/@gyogyszerId",
            document, XPathConstants.NODE);

```

```

csapat.setTextContent("913");

// A Sentor nevű gyógyszer árának módosítása '50000'-re
Node alkalmGyogyszer = (Node)
xpath.evaluate("//alkalmGyogyszer[./megnevezes='Sentor']/ar", document,
    XPathConstants.NODE);

int ar=Integer.parseInt(alkalmGyogyszer.getTextContent());

ar = 50000;

alkalmGyogyszer.setTextContent(Integer.toString(ar));

}

}

private static class DomTraverser {

    public static void traverseLevel(TreeWalker walker, String indent) {

        // Menti a jelenlegi csomópontot
        Node node = walker.getCurrentNode();

        // Kiíratjuk
        if (node.getNodeType() == Node.ELEMENT_NODE) {
            printElementNode(node, indent);
        } else {
            printTextNode(node, indent);
        }

        for (Node n = walker.firstChild(); n != null; n = walker.nextSibling()) {
            traverseLevel(walker, indent + "  ");
        }
    }
}

```

```

        walker.setCurrentNode(node);
    }

    private static void printElementNode(Node node, String indent) {
        System.out.print(indent + node.getNodeName());

        printElementAttributes(node.getAttributes());
    }

    private static void printElementAttributes(NamedNodeMap attributes) {
        int length = attributes.getLength();

        // Attributum(ok) kiírása
        if (length > 0) {
            System.out.print(" [ ");

            for (int i = 0; i < length; i++) {
                Node attribute = attributes.item(i);

                System.out.printf("%s=%s%s", attribute.getNodeName(), attribute.getNodeValue(),
                    i != length - 1 ? ", " : "");
            }

            System.out.println(" ]");
        } else {
            System.out.println();
        }
    }

    private static void printTextNode(Node node, String indent) {
        String content_trimmed = node.getTextContent().trim();

        if (content_trimmed.length() > 0) {

```

```
        System.out.print(indent);

        System.out.printf("{ %s }%n", content_trimmed);
    }
}
}
}
```