

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Állatmenhelyeket tároló adatbázis rendszer

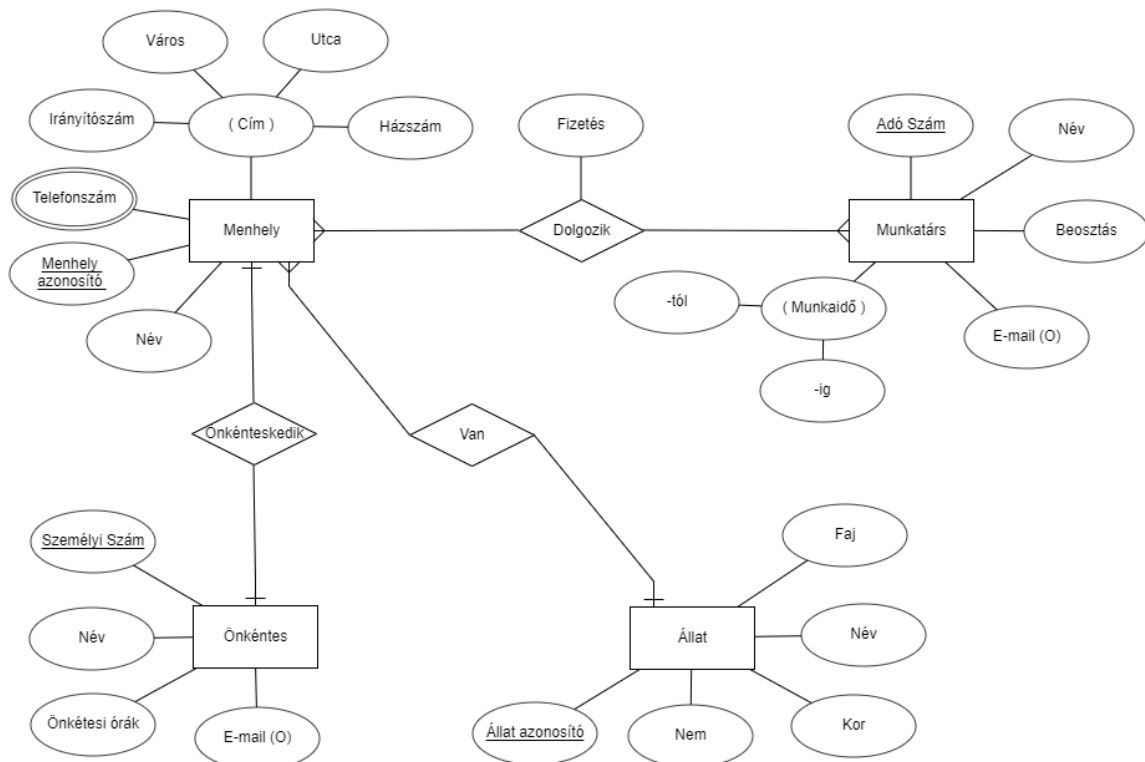
Készítette: **Papp Marcell**

A feladat leírása:

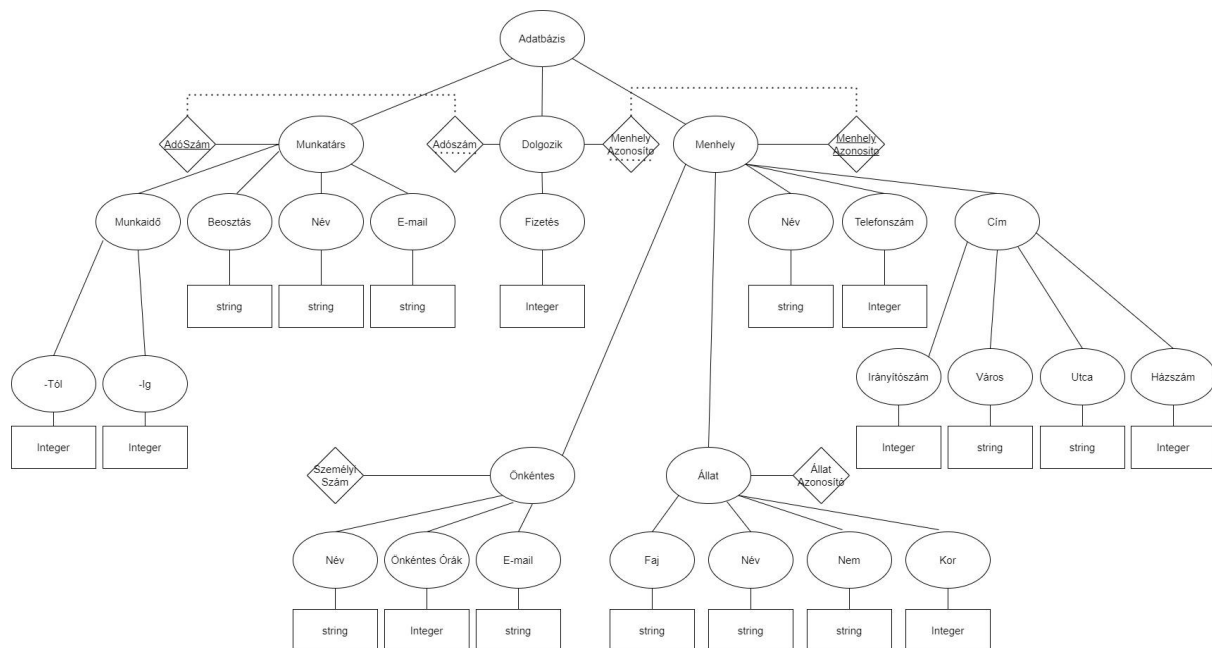
A feladatban egy Állatmenhelyeket nyilvántartó adatbázist tervezek meg. Az adatbázis elemei a Menhelyek, Munkatársak, Önkéntesek, és Állatok. A Menhelyek és az Önkéntesek, illetve az Állatok között 1-M kapcsolat, a Menhely és a Munkatársak között pedig N:M kapcsolat áll fenn. Az N:M kapcsolatnak van 1 "fizetés" tulajdonsága. Az egyedeknek legalább 4 tulajdonsága van, többek között többértékű, opcionális, és összetett tulajdonságok is vannak.

1.Feladat

1a) ER modell



1b) XDM modell



Az 1:több kapcsolatokból gyerekelemek lesznek, a több:több kapcsolatból kapcsoló elem lesz, kulcs tulajdonságokból attribútumok, egyéb tulajdonságokból gyerekelemek lesznek.

1c) XML

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

```
<Adatbázis xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="XMLSchemaX7A7XR.xsd">
```

```
<Menhely M_azonosito="m1">
  <Nev>Kutya Menhely</Nev>
  <Telefonszam>1111</Telefonszam>
  <Telefonszam>2222</Telefonszam>
  <Cim>
    <Iranyitoszam>3500</Iranyitoszam>
    <Varos>Miskolc</Varos>
    <Utca>Főutca</Utca>
    <Hazszam>1</Hazszam>
  </Cim>

  <Onkentes Sz_szam="sz1">
    <Nev>Tamas</Nev>
    <OnkentesOrak>11</OnkentesOrak>
    <Email>Tamas@email.com</Email>
  </Onkentes>
```

<Allat A_azonosito="a1">
 <Nev>Rex</Nev>
 <Faj>Kutya</Faj>
 <Nem>Fiú</Nem>
 <Kor>3</Kor>
</Allat>

<Allat A_azonosito="a2">
 <Nev>Cica</Nev>
 <Faj>Macska</Faj>
 <Nem>Fiú</Nem>
 <Kor>4</Kor>
</Allat>

</Menhely>

<Munkatars Ado_szam="ad1">
 <Nev>Laci</Nev>
 <Email>Laci@email.com</Email>
 <Beosztas>Igazgato</Beosztas>
 <Munkaido>
 <Tol>8</Tol>
 <Ig>16</Ig>
 </Munkaido>
</Munkatars>

<Munkatars Ado_szam="ad2">
 <Nev>Pali</Nev>
 <Email>Pali@email.com</Email>
 <Beosztas>Gondnok</Beosztas>
 <Munkaido>
 <Tol>8</Tol>
 <Ig>16</Ig>
 </Munkaido>
</Munkatars>

<Dolgozik M_ref="m1" A_ref="ad1">
 <Fizetes>100000</Fizetes>
</Dolgozik>

<Dolgozik M_ref="m1" A_ref="ad2">
 <Fizetes>200000</Fizetes>
</Dolgozik>

</Adatbazis>

1d) xml schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Adatbazis" type="adatbazis_tipus">

    <xs:key name="munkatars_key">
      <xs:selector xpath="Munkatars"></xs:selector>
      <xs:field xpath="@Ado_szam"></xs:field>
    </xs:key>

    <xs:key name="menhely_key">
      <xs:selector xpath="Menhely"></xs:selector>
      <xs:field xpath="@M_azonosito"></xs:field>
    </xs:key>

    <xs:key name="onkentes_key">
      <xs:selector xpath="Menhely/Onkentes"></xs:selector>
      <xs:field xpath="@Sz_szam"></xs:field>
    </xs:key>

    <xs:key name="allat_key">
      <xs:selector xpath="Menhely/Allat"></xs:selector>
      <xs:field xpath="@A_azonosito"></xs:field>
    </xs:key>

    <xs:keyref name="dolgozikM" refer="menhely_key">
      <xs:selector xpath="Dolgozik"></xs:selector>
      <xs:field xpath="@M_ref"></xs:field>
    </xs:keyref>

    <xs:keyref name="dolgozikA" refer="munkatars_key">
      <xs:selector xpath="Dolgozik"></xs:selector>
      <xs:field xpath="@A_ref"></xs:field>
    </xs:keyref>

  </xs:element>
  <xs:complexType name="adatbazis_tipus">
    <xs:sequence>
      <xs:element name="Menhely" type="Menhely_tipus"
maxOccurs="unbounded"/>
      <xs:element name="Munkatars" type="Munkatars_tipus"
maxOccurs="unbounded"/>
      <xs:element name="Dolgozik" type="Dolgozik_tipus"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

```

<xs:complexType name="Munkatars_tipus">
  <xs:sequence>
    <xs:element name="Nev" type="xs:string"/>
    <xs:element name="Email" type="xs:string" minOccurs="0"/>
    <xs:element name="Beosztas" type="xs:string"/>
    <xs:element name="Munkaido" type="Munkaido_tipus"/>
  </xs:sequence>
  <xs:attribute name="Ado_szam" type="xs:string" use="required"/>
</xs:complexType>

<xs:complexType name="Munkaido_tipus">
  <xs:sequence>
    <xs:element name="Tol" type="xs:positiveInteger"/>
    <xs:element name="Ig" type="xs:positiveInteger"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="Allat_tipus">
  <xs:sequence>
    <xs:element name="Nev" type="xs:string"/>
    <xs:element name="Faj" type="xs:string"/>
    <xs:element name="Nem" type="xs:string"/>
    <xs:element name="Kor" type="xs:positiveInteger"/>
  </xs:sequence>
  <xs:attribute name="A_azonosito" type="xs:string" use="required"/>
</xs:complexType>

<xs:complexType name="Onkentes_tipus">
  <xs:sequence>
    <xs:element name="Nev" type="xs:string"/>
    <xs:element name="OnkentesOrak" type="xs:positiveInteger"/>
    <xs:element name="Email" type="xs:string" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="Sz_szam" type="xs:string" use="required"/>
</xs:complexType>

<xs:complexType name="Menhely_tipus">
  <xs:sequence>
    <xs:element name="Nev" type="xs:string"/>
    <xs:element name="Telefonszam" type="xs:positiveInteger"
maxOccurs="unbounded"/>
    <xs:element name="Cim" type="Cim_tipus"/>
    <xs:element name="Onkentes" type="Onkentes_tipus"
maxOccurs="1"/>
    <xs:element name="Allat" type="Allat_tipus"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="M_azonosito" type="xs:string" use="required"/>

```

```

</xs:complexType>

<xs:complexType name="Cim_tipus">
  <xs:sequence>
    <xs:element name="Iranditoszam" type="xs:positiveInteger"/>
    <xs:element name="Varos" type="xs:string"/>
    <xs:element name="Utca" type="xs:string"/>
    <xs:element name="Hazszam" type="xs:positiveInteger"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="Dolgozik_tipus">
  <xs:sequence>
    <xs:element name="Fizetes" type="xs:positiveInteger"/>
  </xs:sequence>
  <xs:attribute name="M_ref" type="xs:string" use="required"/>
  <xs:attribute name="A_ref" type="xs:string" use="required"/>
</xs:complexType>

</xs:schema>

```

2.Feladat

2a) adatolvasás DOMReadX7A7XR.java

```

/*
 *Ez a program szinte bármilyen XML fált beolvas és kiírja a konzolra a fájl faszerkezetét.
 *Ezt egy rekurzív metódussal teszi meg, a végigmegy az összes gyerekelemen, és ha van,
 *akkor attribútumaikon és tartalmukon is.
 */

```

```

package hu.domparse.X7A7XR;

import java.io.File;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;

import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.w3c.dom.NamedNodeMap;
import org.w3c.dom.Attr;

public class DOMReadX7A7XR {

```

```

public static void main(String[] args) {
    try {
        DocumentBuilderFactory factory =
            DocumentBuilderFactory.newInstance();
        DocumentBuilder builder = factory.newDocumentBuilder();
        Document document = builder.parse(new File("XMLX7A7XR.xml"));
        document.getDocumentElement().normalize();

        System.out.println("Root element: " +
document.getDocumentElement().getNodeName());
        printNodes(0,document.getChildNodes());

    }catch(Exception e) {
        e.printStackTrace();
    }
}

//Rekurzívan k   r minden elemet   s tartalmukat ha van
private static void printNodes(int depth, NodeList nodeList) {
    for (int i = 0; i < nodeList.getLength(); i++) {
        if (nodeList.item(i).getNodeType() == Node.ELEMENT_NODE) {
            Element element = (Element) nodeList.item(i);
            String name = element.getNodeName();
            String indent = "";

            for (int indentIndex = 0; indentIndex<depth; indentIndex++) {
                indent += " ";
            }

            if (element.getChildNodes().getLength() > 1) {
                System.out.println(indent + name + " : ");
                printAttributes(depth+1,element.getAttributes());
                printNodes(depth+1,element.getChildNodes());
            }
            else {
                System.out.println(indent + name + " : " +
element.getTextContent());
            }
        }
    }
}

//K   rja az attrib  tumokat, ha vannak
private static void printAttributes(int depth, NamedNodeMap attributes) {

```



```

        String indent = "";

        for (int indentIndex = 0; indentIndex<depth; indentIndex++) {
            indent += " ";
        }

        for (int i = 0; i<attributes.getLength(); i++) {
            Attr attribute = (Attr) attributes.item(i);
            String name = attribute.getNodeName();
            System.out.println(indent + name + " : " + attribute.getNodeValue());
        }

    }

}

```

2b) adatmódosítás DOMModifyX7A7XR.java

```

/*
 * Ez a program az XML fájlmon fog egy specifikus változtatást elvégezni, majd kiírni a
 * konzolra.
 * A változtatás tárgya az "m1" azonosítójú menhely nevének (alapból "Kutya Menhely")
 * megváltoztatása "Modositott nev"-re. Működése során kigyűjti a menhely objektumokat a
 * DOMból,
 * majd egy for ciklussal megkeresi az illeszkedő azonosítójút, megváltoztatja a nevét,
 * majd az egész XML fájlt kiírja konzolra a TransformerFactory segítségével.
 */

```

```

package hu.domparse.X7A7XR;

```

```

import java.io.File;

```

```

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
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 DOMModifyX7A7XR {

```

```

public static void main(String[] args) {
    try {
        DocumentBuilderFactory factory =
            DocumentBuilderFactory.newInstance();
        DocumentBuilder builder = factory.newDocumentBuilder();
        Document document = builder.parse(new File("XMLX7A7XR.xml"));
        document.getDocumentElement().normalize();

        //m1 azonosítójú menhely nevének megváltoztatása

        NodeList shelters=document.getElementsByTagName("Menhely");
        for(int i=0;i<shelters.getLength();i++) {
            if(shelters.item(i).getNodeType()==Node.ELEMENT_NODE) {
                Element element = (Element) shelters.item(i);

                //System.out.println(element.getAttribute("M_azonosito"));
                if(element.getAttribute("M_azonosito").equals("m1")) {

                    element.getElementsByTagName("Nev").item(0).setTextContent("Modositott nev");
                }
            }
        }

        // Kiírás konzolra
        TransformerFactory transformerFactory =
            TransformerFactory.newInstance();
        Transformer transformer = transformerFactory.newTransformer();
        DOMSource source = new DOMSource(document);
        System.out.println("-----Modified File-----");
        StreamResult consoleResult = new StreamResult(System.out);
        transformer.transform(source, consoleResult);

        }catch(Exception e) {
            e.printStackTrace();
        }
    }
}

```

2c adatlekérdezés DOMQueryX7A7XR.java

/*

* Ez a program egy lekérdezést hajt végre az XML fájlmon, majd kiírja a lekérdezés eredményét

* a konzolra. A lekérdezés célja azoknak az önkénteseknek a nevei akik legalább 11-órát önkénteskedtek.

```
* A program kigyűjti a önkéntes objektumokat a DOMból, majd egy for ciklussal megkeresi  
* a kritériumnak megfelelő elemeket, és kiírja a nevüket a konzolra  
*/
```

```
package hu.domparse.X7A7XR;
```

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

```
import javax.xml.parsers.DocumentBuilder;  
import javax.xml.parsers.DocumentBuilderFactory;
```

```
import org.w3c.dom.Document;  
import org.w3c.dom.Element;  
import org.w3c.dom.Node;  
import org.w3c.dom.NodeList;
```

```
public class DOMQueryX7A7XR {
```

```
    public static void main(String[] args) {  
        try {
```

```
            DocumentBuilderFactory factory =  
                DocumentBuilderFactory.newInstance();  
            DocumentBuilder builder = factory.newDocumentBuilder();  
            Document document = builder.parse(new File("XMLX7A7XR.xml"));  
            document.getDocumentElement().normalize();
```

```
            //Query azoknak az önkénteseknek a nevei akik legalább 11-órát
```

```
önkénteskedtek
```

```
            NodeList volunteers=document.getElementsByTagName("Onkentes");  
            for(int i=0;i<volunteers.getLength();i++) {  
                if(volunteers.item(i).getNodeType()!=Node.ELEMENT_NODE)
```

```
{
```

```
                Element element = (Element) volunteers.item(i);
```

```
                if(Integer.parseInt(element.getElementsByTagName("OnkentesOrak").item(0).getTextContent())>=11) {
```

```
                    System.out.println(element.getElementsByTagName("Nev").item(0).getTextContent());  
                }
```

```
            }  
        }
```

```
    }catch(Exception e) {  
        e.printStackTrace();  
    }
```

}

}