

Jegyzőkönyv

Adatkezelés XML környezetben

Féléves feladat



Készítette: Jakab Alexandra

Neptun kód: ZVC5K4

Gyakorlat: Kedd 14⁰⁰ – 16⁰⁰

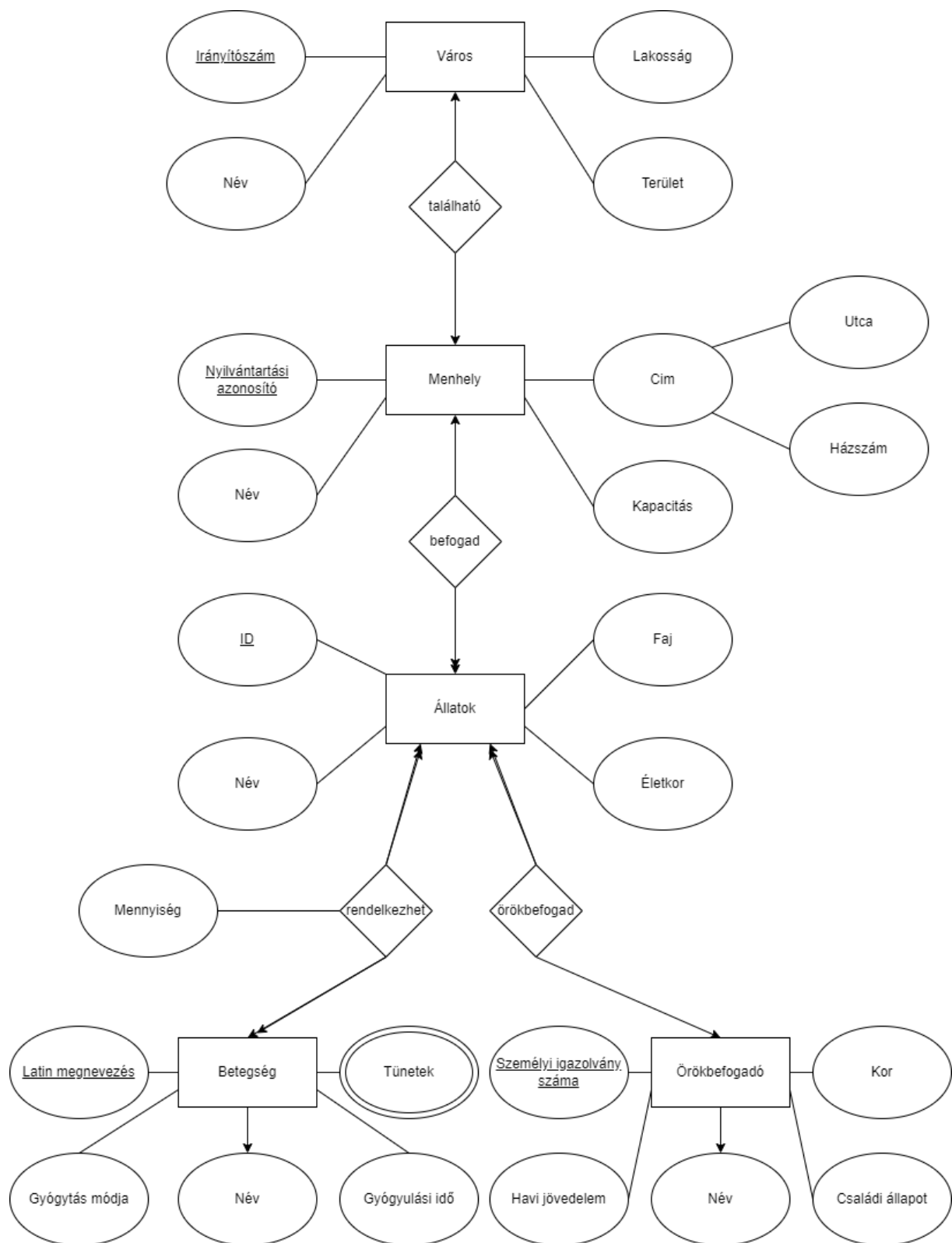
Gyakorlatvezető: Dr. Bednarik László

2021.11.29. **Miskolc**

A feladat:

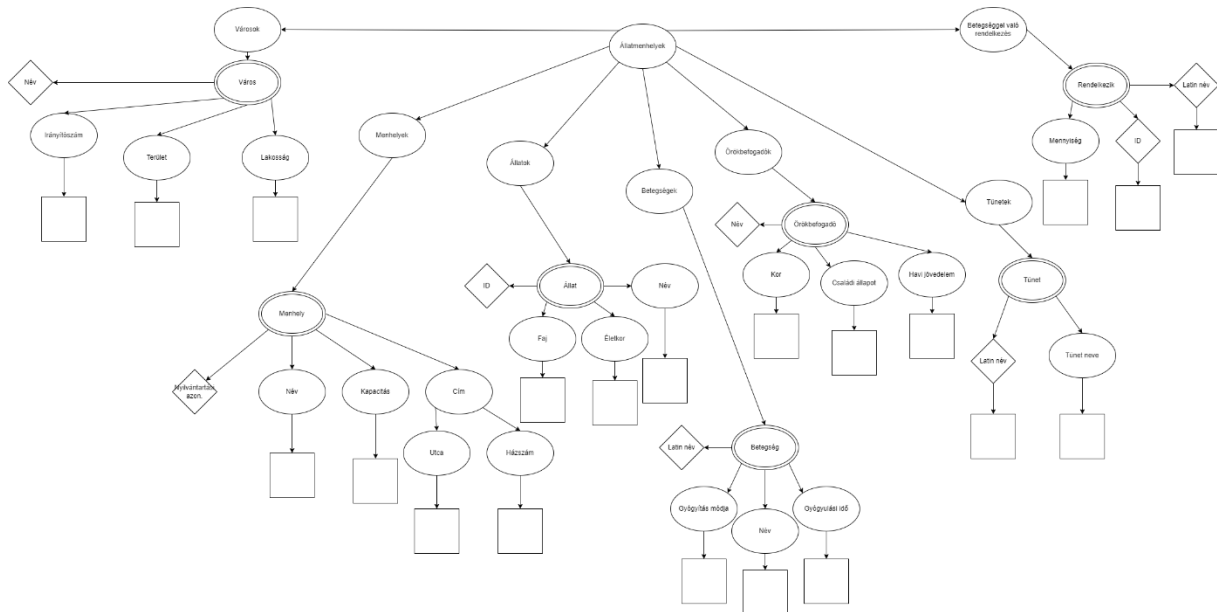
A feladat témája, hogy bemutassa egy állatmenhely adatait, benne megtalálható állatok, illetve az őket örökbefogadó személyeket mindegy egyes tulajdonságukkal hogyan tudunk eltárolni egy adatbázisban. Egy menhely értelemszerűen egy városban található, ezért azokhoz tartozó adatokat is számba kell venni. Az ehhez tartozó tulajdonságok, pedig a város irányítószáma, neve, lakossági adatai, illetve területe km^2 -ben megadva. Mivel menhelyekről beszélünk, ezért azokhoz is tartoznak tulajdonságok, egy nyilvántartási azonosító, a menhely neve, az állatokat befogadó kapacitása, illetve a címe, ami egy összetett tulajdonság, ezért ez egy utcából és egy házszámból áll össze. A feladatban egy városban csak egy menhely létezik, ám ez valóságban nem feltétlen igaz. A különböző menhelyek állatokat fogadnak be, melyek szintén rendelkeznek tulajdonságokkal. Ezek a tulajdonságok az ID, az állat neve, amelyet a menhelyen választanak ki neki, az állat fajtája, illetve annak az életkora. Egy menhelyen több állat is élhet, azonban egy állat több menhelyen nem. Az állatok sajnos betegen is kerülhetnek menhelyre és rendelkezhetnek több fajta betegséggel, - ezért a kapcsolatuknak is van egy mennyiség tulajdonsága-, illetve több állatnak is lehet ugyan olyan betegsége. Ezeket a betegségek szintén rendelkeznek tulajdonságokkal, amely a latin megnevezése, a magyar neve, a gyógyítás módja, a gyógyulásai idő, illetve a tünetei. A tünetek egy többértékű tulajdonság, mivel egy betegséghez több tünetegyüttes is tartozhat. A különböző állatokat, ha szerencsések akkor örökbe fogadják, ezért számításba kell venni az örökbefogadókat is, akik az előbb említettekhez hasonlóan rendelkeznek tulajdonságokkal. Ezek a tulajdonságok a személyi igazolvány szám, az örökbefogadó neve, a kora, ezen felül a havi jövedelme és a családi állapot. Az utolsó kettő kiemelten fontos, mivel érdemes úgy állatot vállalni, hogy van hozzá elegendő jövedelme az illetőnek, illetve az is fontos, hogy a kiválasztott állat mennyire van hozzászokva az emberekhez. Ez annyit jelent, hogy ha nem feltételül a legemberbarátibb állatról beszélünk, akkor nem biztos, hogy megfelel neki egy olyan környezet, ahol egy nagy család veszi körül.

Az adatbázis ER modellje:



Az ER modell konvertálása XDM modellre:

XDM modellnél háromféle jelölés alkalmazunk. Ezek az ellipszis, a rombusz, illetve a téglalap. Az ellipszis jelöli az elemeket, minden egyedből elem lesz, ezen felül a tulajdonságokból is. A rombusz jelöli az attribútumokat, amelyek a kulcs tulajdonságokból keletkeznek. A téglalap jelöli a szöveget, amely majd az XML dokumentumban fog megjelenni. Azoknak az elemeknek, amelyek többször is előfordulhatnak, a jelölése dupla ellipszissel történik.



XML dokumentum készítés az XDM modell alapján:

```
<?xml version="1.0" encoding="UTF-8" ?>
<allatmenhelyek>
  <varosok>
    <varos iranyitoszam="3515">
      <nev>Miskolc</nev>
      <terulet>236.7</terulet>
      <lakossag>157166</lakossag>
    </varos>
    <varos iranyitoszam="4400">
      <nev>Nyíregyháza</nev>
      <terulet>274.5</terulet>
      <lakossag>119132</lakossag>
    </varos>
  </varosok>
  <menhelyek>
    <menhely nyilvantartasi_azon="237612" iranyitoszam ="3515">
      <nev>Miskolci Állatsegítő Alapítvány</nev>
      <cim>
        <utca>Szigeti</utca>
        <hazszam>2</hazszam>
      </cim>
      <kapacitas>34100</kapacitas>
    </menhely>
    <menhely nyilvantartasi azon="452891" iranyitoszam="4400">
```

```

    <nev>Állatbarát Alapítvány </nev>
    <cim>
        <utca>Csatorna</utca>
        <hazszam>2</hazszam>
    </cim>
    <kapacitas>18200</kapacitas>
</menhely>
</menhelyek>

<allatok>
    <allat id ="1" nyilvantartasi_azon="237612" sz_szam="987654sd">
        <nev>Bodri</nev>
        <faj>Kutya</faj>
        <eletkor>1</eletkor>
    </allat>
    <allat id ="2" nyilvantartasi_azon="452891" sz_szam="946234hu">
        <nev>Nyafi</nev>
        <faj>Macska</faj>
        <eletkor>3</eletkor>
    </allat>
    <allat id ="3" nyilvantartasi_azon="452891" sz_szam="234165er">
        <nev>Wiskey</nev>
        <faj>Kutya</faj>
        <eletkor>9</eletkor>
    </allat>
    <allat id ="4" nyilvantartasi_azon="237612" sz_szam="987654sd">
        <nev>Peti</nev>
        <faj>Papagáj</faj>
        <eletkor>1</eletkor>
    </allat>
    <allat id ="5" nyilvantartasi_azon="452891" sz_szam="123456as">
        <nev>Töpszli</nev>
        <faj>Nyúl</faj>
        <eletkor>2</eletkor>
    </allat>
</allatok>

<betegsegek>
    <betegseg latin_nev="myxomatosis" >
        <nev>Myxomatosis</nev>
        <gyogvyitas_modja>Védőoltás</gyogvyitas_modja>
        <gyogyulasi_ido>14 nap</gyogyulasi_ido>
    </betegseg>
    <betegseg latin_nev="canine_parvovirus">
        <nev>Parvovírusos bélgyulladás</nev>
        <gyogvyitas_modja>Védőoltás</gyogvyitas_modja>
        <gyogyulasi_ido>6 nap</gyogyulasi_ido>
    </betegseg>
    <betegseg latin_nev="feline_infectious_peritonitis">
        <nev>Fertőző hashártyagyulladás (FIP)</nev>
        <gyogvyitas_modja>Antibiotikumos kezelés</gyogvyitas_modja>
        <gyogyulasi_ido>16</gyogyulasi_ido>
    </betegseg>
</betegsegek>

<betegseggel_rendelkezes>
    <rendelkezik id="1" latin_nev="canine_parvovirus">
        <mennyiseg>1</mennyiseg>

```

```
</rendelkezik>
<rendelkezik id="2" latin_nev="feline_infectious_peritonitis">
  <mennyiseg>1</mennyiseg>
</rendelkezik>
<rendelkezik id="3" latin_nev="canine_parvovirus">
  <mennyiseg>2</mennyiseg>
</rendelkezik>
<rendelkezik id="3" latin_nev="myxomatosis">
  <mennyiseg>2</mennyiseg>
</rendelkezik>
<rendelkezik id="5" latin_nev="myxomatosis">
  <mennyiseg>1</mennyiseg>
</rendelkezik>
</betegseggel_rendelkezes>
```

```
<tunetek>
  <tunet latin_nev="myxomatosis">
    <tunet_neve>Fül duzzanat</tunet_neve>
  </tunet>
  <tunet latin_nev="myxomatosis">
    <tunet_neve>Orr duzzanat</tunet_neve>
  </tunet>
  <tunet latin_nev="myxomatosis">
    <tunet_neve>Orrfolyás</tunet_neve>
  </tunet>
  <tunet latin_nev="canine_parvovirus">
    <tunet_neve>Láz</tunet_neve>
  </tunet>
  <tunet latin_nev="canine_parvovirus">
    <tunet_neve>Hányás</tunet_neve>
  </tunet>
  <tunet latin_nev="canine_parvovirus">
    <tunet_neve>Hasmenés</tunet_neve>
  </tunet>
  <tunet latin_nev="feline_infectious_peritonitis">
    <tunet_neve>Koordinációs zavar</tunet_neve>
  </tunet>
  <tunet latin_nev="feline_infectious_peritonitis">
    <tunet_neve>Görcsök</tunet_neve>
  </tunet>
  <tunet latin_nev="feline_infectious_peritonitis">
    <tunet_neve>Veseelégtelenség</tunet_neve>
  </tunet>
</tunetek>
```

```
<orokbefogadok>
  <orokbefogado sz_szam="987654sd">
    <nev>Kiss Béla</nev>
    <kor>25</kor>
    <havi_jovedelem>340000</havi_jovedelem>
    <csaladi_allapot>házas</csaladi_allapot>
  </orokbefogado>
  <orokbefogado sz_szam="123456as">
    <nev>Pélida Katalin</nev>
    <kor>18</kor>
    <havi_jovedelem>150000</havi_jovedelem>
    <csaladi_allapot>egyedülálló</csaladi_allapot>
  </orokbefogado>
```

```

<orokbefogado sz_szam="234165er">
  <nev>Nagy Ferenc</nev>
  <kor>67</kor>
  <havi_jovedelem>120000</havi_jovedelem>
  <csaladi_allapot>özvegy</csaladi_allapot>
</orokbefogado>
<orokbefogado sz_szam="946234hu">
  <nev>Lakatos Beáta</nev>
  <kor>44</kor>
  <havi_jovedelem>500000</havi_jovedelem>
  <csaladi_allapot>házas</csaladi_allapot>
</orokbefogado>
</orokbefogadok>
</allatmenhelyek>

```

XMLSchema készítése az XML dokumentum alapján:

```

<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="allatmenhelyek"/>
  <xs:element name="varosok">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="varos">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="nev" type="xs:string"/>
              <xs:element name="terulet" type="xs:float"/>
              <xs:element name="lakossag" type="xs:positiveInteger"/>
            </xs:sequence>
            <xs:attribute name="iranyitoszam" type="xs:integer"/>
          </xs:complexType>
          <xs:key name="varosKulcs">
            <xs:selector xpath="varos"></xs:selector>
            <xs:field xpath="iranyitoszam"></xs:field>
          </xs:key>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="menhelyek">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="menhely">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="nev" type="xs:string"/>
              <xs:element name="cim">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="utca" type="xs:string"/>
                    <xs:element name="hazszam"
type="hazszamTipus"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```

```

        </xs:complexType>
        </xs:element>
        <xs:element name="kapacitas"
type="xs:positiveInteger"/>
    </xs:sequence>
    <xs:attribute name="nyilvantartas_azon" type="xs:integer"/>
    <xs:attribute name="iranyitoszam" type="xs:integer"/>
</xs:complexType>
<xs:key name="menhelyKulcs">
    <xs:selector xpath="menhely"></xs:selector>
    <xs:field xpath="nyilvantartas_azon"></xs:field>
</xs:key>
<xs:keyref name="varosRef" refer="varosKulcs">
    <xs:selector xpath="menhely"></xs:selector>
    <xs:field xpath="@iranyitoszam"></xs:field>
</xs:keyref>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="allatok">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="allat">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="nev" type="xs:string"/>
                        <xs:element name="faj" type="fajTipus"/>
                        <xs:element name="eletkor" type="xs:integer"/>
                    </xs:sequence>
                    <xs:attribute name="id" type="xs:integer"/>
                    <xs:attribute name="nyilvantartasi_azon"
type="xs:integer"/>
                    <xs:attribute name="sz_szam" type="szSzamTipus"/>
                </xs:complexType>
                <xs:key name="allatKulcs">
                    <xs:selector xpath="allat"></xs:selector>
                    <xs:field xpath="id"></xs:field>
                </xs:key>
                <xs:keyref name="menhelyRef" refer="menhelyKulcs">
                    <xs:selector xpath="allat"></xs:selector>
                    <xs:field xpath="@nyilvantartasi_azon"></xs:field>
                </xs:keyref>
                <xs:keyref name="orokbefogadoRef" refer="orogbefogadoKulcs">
                    <xs:selector xpath="allat"></xs:selector>
                    <xs:field xpath="@sz_szam"></xs:field>
                </xs:keyref>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="betegsegek">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="betegseg">
                <xs:complexType>
                    <xs:sequence>

```



```

        <xs:element name="nev" type="xs:string"/>
        <xs:element name="gyogyitas_modja" type="xs:string"/>
        <xs:element name="gyogyulasi_ido"
type="xs:positiveInteger"/>
    </xs:sequence>
    <xs:attribute name="latin_nev" type="xs:string"/>
    <xs:attribute name="id" type="xs:integer"/>
</xs:complexType>
<xs:key name="betegsegKulcs">
    <xs:selector xpath="betegseg"></xs:selector>
    <xs:field xpath="latin_nev"></xs:field>
</xs:key>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="betegseggel_rendelkezes">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="rendelkezik">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="mennyiseg"/>
                    </xs:sequence>
                    <xs:attribute name="id" type="xs:integer"/>
                    <xs:attribute name="latin_nev" type="xs:string"/>
                </xs:complexType>
                <xs:keyref name="rendelkezésKapcsAllat" refer="allatKulcs">
                    <xs:selector xpath="rendelkezik"></xs:selector>
                    <xs:field xpath="@idv"></xs:field>
                </xs:keyref>
                <xs:keyref name="rendelkezésKapcsBetegseg"
refer="betegsegKulcs">
                    <xs:selector xpath="rendelkezik"></xs:selector>
                    <xs:field xpath="@latin_nev"></xs:field>
                </xs:keyref>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="tunetek">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="tunet">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="tunet_neve" type="xs:string"/>
                    </xs:sequence>
                    <xs:attribute name="latin_nev" type="xs:string"/>
                </xs:complexType>
                <xs:keyref name="tunetRef" refer="betegsgKulcs">
                    <xs:selector xpath="tunet"></xs:selector>
                    <xs:field xpath="@latin_nev"></xs:field>
                </xs:keyref>
            </xs:element>
        </xs:sequence>
    </xs:complexType>

```

```

</xs:element>

<xs:element name="orokbefogadok">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="orokbefogado">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="nev" type="xs:string"/>
            <xs:element name="kor" type="korTipus"/>
            <xs:element name="havi_jovedelem"
type="xs:positiveInteger"/>
            <xs:element name="csaladi_allapot"
type="csaladiAllapotTipus"/>
          </xs:sequence>
          <xs:attribute name="sz_szam" type="szSzamTipus"/>
        </xs:complexType>
        <xs:key name="orokbefogadoKulcs">
          <xs:selector xpath="orokbefogado"></xs:selector>
          <xs:field xpath="sz_szam"></xs:field>
        </xs:key>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:simpleType name="hazszamTipus">
  <xs:restriction base="xs:positiveInteger">
    <xs:minInclusive value="1"/>
    <xs:maxInclusive value="200"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="fajTipus">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Macska"/>
    <xs:enumeration value="Kutya"/>
    <xs:enumeration value="Nyúl"/>
    <xs:enumeration value="Papagáj"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="szSzamTipus">
  <xs:restriction base="xs:string">
    <xs:pattern value="[1-9]{1} [0-9] {5} [a-z]{2}"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="korTipus">
  <xs:restriction base="xs:positiveInteger">
    <xs:minInclusive value="18"/>
    <xs:maxInclusive value="70"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="csaladiAllapotTipus">
  <xs:restriction base="xs:string">

```

```

        <xs:enumeration value="házas"/>
        <xs:enumeration value="egyedülálló"/>
        <xs:enumeration value="elvált"/>
        <xs:enumeration value="özvegy"/>
    </xs:restriction>
</xs:simpleType>
</xs:schema>

```

DOM program készítése az XML dokumentum adatainak:

Adatok beolvasása:

```

package hu.domparse.zvc5k4;

import java.io.File;
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;

public class DOMReadZVC5K4 {

    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("XMLZVC5K4.xml"));
            document.getDocumentElement().normalize();
            System.out.println("Gyökér elem: " +
document.getDocumentElement().getNodeName());
            System.out.println("\n");

            //Az elem meghatározása
            list = document.getElementsByTagName("varos");
            for(int i = 0; i < list.getLength(); i++){
                Node node = list.item(i);
                //Aktuális elem
                System.out.println("Jelenleg ennek az elemnek a tulajdonságait
lehet látni: " + node.getNodeName());

                //Város adatainak kiíratása
                if (node.getNodeType() == Node.ELEMENT_NODE){
                    Element element = (Element) node;
                    System.out.println("Irányítószám" +
element.getAttribute("iranyitoszam"));

                    System.out.println("A város neve: " +
element.getElementsByTagName("nev").item(0).getTextContent());
                    System.out.println("A város területe: " +
element.getElementsByTagName("terulet").item(0).getTextContent());
                    System.out.println("A város lakossága: " +
element.getElementsByTagName("lakosság").item(0).getTextContent());
                }
            }
        }
    }
}

```

```

    }

    //Az elem meghatározása
    list = document.getElementsByTagName("menhely");
    for(int i = 0; i < list.getLength(); i++){
        Node node = list.item(i);
        //Aktuális elem
        System.out.println("Jelenleg ennek az elemnek a tulajdonságait
lehet látni: " + node.getNodeName());

        //Menhely adatainak kiíratása
        if (node.getNodeType() == Node.ELEMENT_NODE){
            Element element = (Element) node;
            System.out.println("Menhely nyilvántartási azonosítója: " +
element.getAttribute("nyilvantart_azon"));

            System.out.println("A menhely neve: " +
element.getElementsByTagName("nev").item(0).getTextContent());
            System.out.println("A menhely kapacitása: " +
element.getElementsByTagName("kapacitas").item(0).getTextContent());

            Node lakcim = list.item(i);
            if(lakcim.getNodeType() == Node.ELEMENT_NODE){
                Element lakcimElement = (Element) node;

                System.out.println("A menhely címe: ");

                System.out.println("Utca: " +
lakcimElement.getElementsByTagName("utca").item(0).getTextContent());
                System.out.println("Hászám: " +
lakcimElement.getElementsByTagName("hazszam").item(0).getTextContent());
            }
        }
    }

    //Az elem meghatározása
    list = document.getElementsByTagName("allat");
    for (int i = 0; i < list.getLength(); i++){
        Node node = list.item(i);
        //Aktuális elem
        System.out.println("Jelenleg ennek az elemnek a tulajdonságait
lehet látni: " + node.getNodeName());

        //Állatok adatainak kiíratása
        if (node.getNodeType() == Node.ELEMENT_NODE){
            Element element = (Element) node;
            System.out.println("ID: " + element.getAttribute("id"));
            System.out.println("A menhely nyilvántartási azonosítója: " +
element.getAttribute("nyilvantartasi_azonS"));
            System.out.println("Az örökbefogadó személyi igazolvány száma:
" + element.getAttribute("sz_szam"));

            System.out.println("Az állat neve: " +
element.getElementsByTagName("nev").item(0).getTextContent());
            System.out.println("Az állat fajtája: " +
element.getElementsByTagName("faj").item(0).getTextContent());
            System.out.println("Az állat kora évben: " +
element.getElementsByTagName("eletkor").item(0).getTextContent());
        }
    }

    //Az elem meghatározása
    list = document.getElementsByTagName("betegseg");
    for (int i = 0; i < list.getLength(); i++){

```

```

        Node node = list.item(i);
        //Aktuális elem
        System.out.println("Jelenleg ennek az elemnek a tulajdonságait
lehet látni: " + node.getNodeName());

        //Betegségek adatainak kiíratása
        if (node.getNodeType() == Node.ELEMENT_NODE) {
            Element element = (Element) node;
            System.out.println("A betegség latin neve: " +
element.getAttribute("latin_nev"));
            System.out.println("Annak az állatnak az id-je amelyik
rendelkezik ezzel a betegséggel: " + element.getAttribute("id"));

            System.out.println("A betegség neve: " +
element.getElementsByTagName("nev").item(0).getTextContent());
            System.out.println("A betegség gyógyítási módja: " +
element.getElementsByTagName("gyogyitas_modja").item(0).getTextContent());
            System.out.println("A betegség gyógyulási ideje napban megadva:
" + element.getElementsByTagName("gyogyulasi_ido").item(0).getTextContent());
        }
    }

    //Az elem meghatározása
    list = document.getElementsByTagName("rendelkezik");
    for (int i = 0; i < list.getLength(); i++){
        Node node = list.item(i);
        //Aktuális elem
        System.out.println("Jelenleg ennek az elemnek a tulajdonságait
lehet látni: " + node.getNodeName());

        //Az állatok betegségeinek mennyiségét adja meg
        if (node.getNodeType() == Node.ELEMENT_NODE) {
            Element element = (Element) node;
            System.out.println("Az állat, amely betegséggel rendelkezik: "
+ element.getAttribute("id"));
            System.out.println("A betegség latin neve: " +
element.getAttribute("latin_nev"));

            System.out.println("A betegségek mennyisége, amellyel a
megadott állat rendelkezik: " +
element.getElementsByTagName("mennyiseg").item(0).getTextContent());
        }
    }

    //Az elem meghatározása
    list = document.getElementsByTagName("tunet");
    for (int i = 0; i < list.getLength(); i++){
        Node node = list.item(i);
        //Aktuális elem
        System.out.println("Jelenleg ennek az elemnek a tulajdonságait
lehet látni: " + node.getNodeName());

        //Tünetek kiíratása
        if (node.getNodeType() == Node.ELEMENT_NODE) {
            Element element = (Element) node;
            System.out.println("A betegség latin neve, amelyhez a tünet
tartozik: " + element.getAttribute("latin_nev"));

            System.out.println("A tünet neve: " +
element.getElementsByTagName("tunet_neve").item(0).getTextContent());
        }
    }

    //Az elem meghatározása

```

```

        list = document.getElementsByTagName("orokbefogado");
        for (int i = 0; i < list.getLength(); i++){
            Node node = list.item(i);
            //Aktuális elem
            System.out.println("Jelenleg ennek az elemnek a tulajdonságait
lehet látni: " + node.getNodeName());

            //Örökbefogadók adatainak kiíratása
            if (node.getNodeType() == Node.ELEMENT_NODE){
                Element element = (Element) node;
                System.out.println("Az örökbefogadó személyi igazolvány száma:
" + element.getAttribute("sz_szam"));

                System.out.println("Az örökbefogadó neve: " +
element.getElementsByTagName("nev").item(0).getTextContent());
                System.out.println("Az örökbefogadó életkora: " +
element.getElementsByTagName("kor").item(0).getTextContent());
                System.out.println("Az örökbefogadó jövedelme: " +
element.getElementsByTagName("havi_jovedelem").item(0).getTextContent());
                System.out.println("Az örökbefogadó családi állapota: " +
element.getElementsByTagName("csaladi_allapot").item(0).getTextContent());
            }
        }

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

```

Adatok lekérdezése:

```

package hu.domparse.zvc5k4;

import java.io.File;
import java.lang.reflect.Array;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.transform.TransformerException;

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

public class DOMQueryZVC5K4 {
    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("XMLZVC5K4.xml"));
            document.getDocumentElement().normalize();

            //Az XML fájl belolvasása és elemzése után a gyökér elem
            System.out.println("Gyökér elem: " +
document.getDocumentElement().getNodeName());

```

```

        System.out.println("\n");

        //Első lekérdezés, ahol az állatok adatait írja ki
        System.out.println("Az eltárolt állatok adatai: \n");
        list = document.getElementsByTagName("allat");
        for (int i = 0; i < list.getLength(); i++){
            Node node = list.item(i);
            System.out.println("Jelenleg ennek az elemnek a tulajdonságait
lehet látni: " + node.getNodeName());
            if (node.getNodeType() == Node.ELEMENT_NODE){
                Element element = (Element) node;
                System.out.println("ID: " + element.getAttribute("id"));
                System.out.println("A menhely nyilvántartási azonosítója: " +
element.getAttribute("nyilvantartasi_azonS"));
                System.out.println("Az örökbefogadó személyi igazolvány száma:
" + element.getAttribute("sz_szam"));

                System.out.println("Az állat neve: " +
element.getElementsByTagName("nev").item(0).getTextContent());
                System.out.println("Az állat fajtája: " +
element.getElementsByTagName("faj").item(0).getTextContent());
                System.out.println("Az állat kora évben: " +
element.getElementsByTagName("eletkor").item(0).getTextContent());
            }
        }
        System.out.println("\n");

        //Második lekérdezés, kilistázza azoknak az örökbefogadónak az adatait,
akik házasok
        System.out.println("Házass örökbefogadók: \n");
        list = document.getElementsByTagName("orokbefogado");
        for (int i = 0; i < list.getLength(); i++){
            Node node = list.item(i);

            if (node.getNodeType() == Node.ELEMENT_NODE){
                Element element = (Element) node;
                Node node1 =
element.getElementsByTagName("csaladi_allapot").item(0);
                String allapot = node1.getTextContent();

                if ("házas".equals(allapot)){
                    System.out.println("Az örökbefogadó személyi igazolvány
száma: " + element.getAttribute("sz_szam"));

                    System.out.println("Az örökbefogadó neve: " +
element.getElementsByTagName("nev").item(0).getTextContent());
                    System.out.println("Az örökbefogadó életkora: " +
element.getElementsByTagName("kor").item(0).getTextContent());
                    System.out.println("Az örökbefogadó jövedelme: " +
element.getElementsByTagName("havi_jovedelem").item(0).getTextContent());
                    System.out.println("Az örökbefogadó családi állapota: " +
element.getElementsByTagName("csaladi_allapot").item(0).getTextContent());
                }
            }
        }
    } catch (ParserConfigurationException e){
        e.printStackTrace();
    } catch (Exception e){
        e.printStackTrace();
    }
}
}

```

Adatok módosítása:

```
package hu.domparse.zvc5k4;

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 DOMModifyZVC5K4 {

    public static void main(String[] args){
        NodeList list;

        try{
            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
            DocumentBuilder builder = factory.newDocumentBuilder();
            //Fájl beolvasás
            Document document = builder.parse(new File("XMLZVC5K4.xml"));
            document.getDocumentElement().normalize();

            //Elem meghatározása
            list = document.getElementsByTagName("betegseg");
            for(int i = 0; i < list.getLength(); i++){
                Node node = list.item(i);
                //Aktuális elem
                System.out.println("Jelenleg ennek az elemnek a tulajdonságait
lehet látni: " + node.getNodeName());

                //Betegségek adatainak kiíratása
                if (node.getNodeType() == Node.ELEMENT_NODE){
                    Element element = (Element) node;

                    String latin_nev = element.getAttribute("latin_nev");

                    System.out.println("A betegség latin neve: " +
element.getAttribute("latin_nev"));
                    System.out.println("Annak az állatnak az id-je amelyik
rendelkezik ezzel a betegséggel: " + element.getAttribute("id"));

                    System.out.println("A betegség neve: " +
element.getElementsByTagName("nev").item(0).getTextContent());
                    System.out.println("A betegség gyógyítási módja: " +
element.getElementsByTagName("gyogyitas_modja").item(0).getTextContent());

                    //Gyógyulási idő módosítása a myxomatosis nevű betegségnél
                    if(latin_nev.equals("myxomatosis")){
                        element.getElementsByTagName("gyogyulasi_ido").item(0).setTextContent("20");
                    }
                }
            }
        }
    }
}
```



```

        System.out.println("A betegség gyógyulási ideje napban megadva:
" + element.getElementsByTagName("gyogyulasi_ido").item(0).getTextContent());
    }
}

//Adatok lekérdezése, illetve kiírása egy xml fájlba
Transformer transformer =
TransformerFactory.newInstance().newTransformer();
Source input = new DOMSource(document);
Result output = new StreamResult(new File("XMLZVC5K4Modify.xml"));
transformer.transform(input, output);

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

}

```