

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Horgász Verseny

Készítette: **Schmidt Bence**

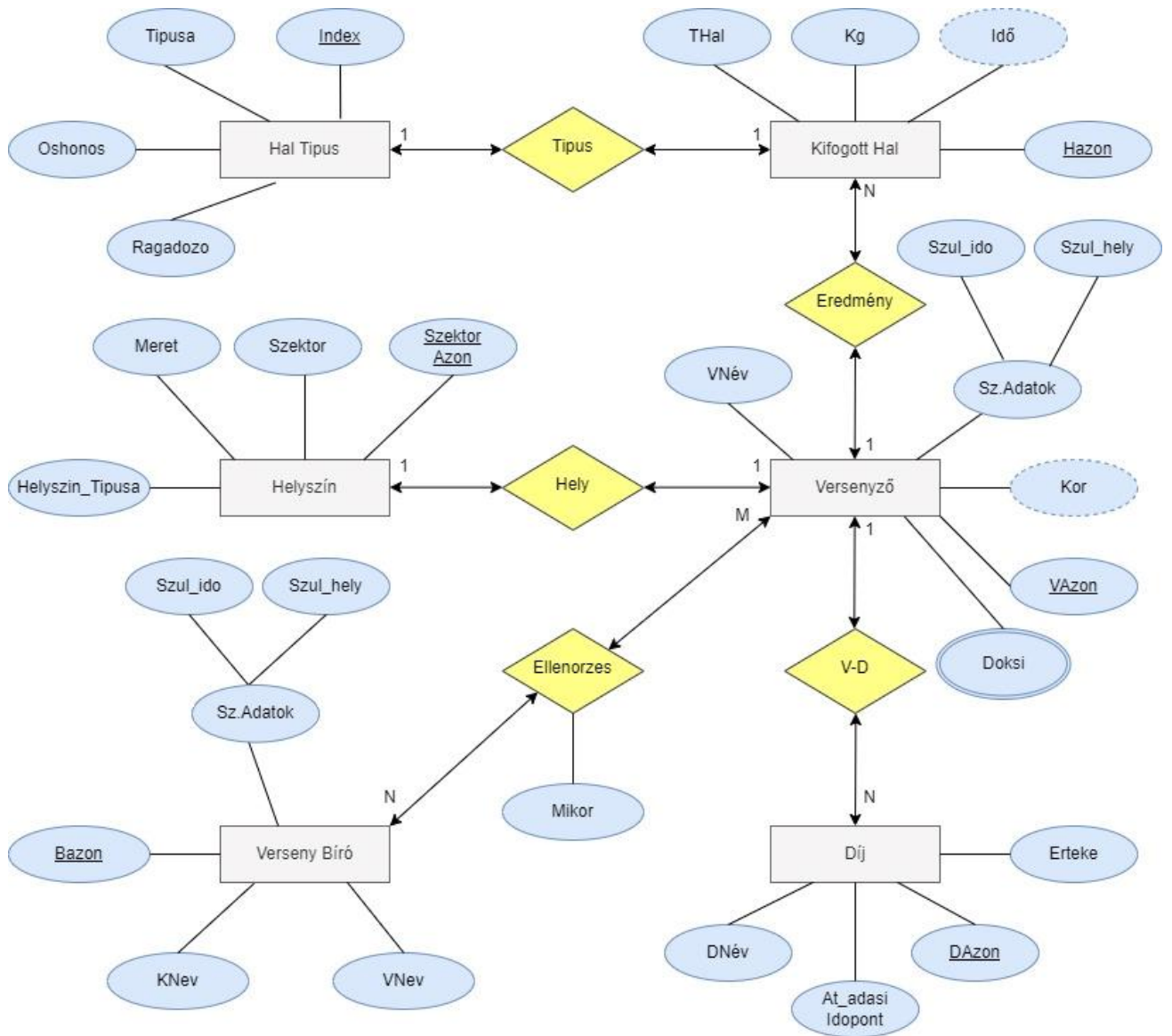
Neptunkód: **D1HKJZ**

A feladat leírása:

Feladatom egy olyan adatbázis ER modelljének a megtervezése, ami megfelel a normálformáknak és nem tartalmaz redundanciát. Az adatbázis megfelelő lesz a horgászversenyen történt dolgok tárolására és az eredmények lekérdezésére akár az adott kifogott hal fajtának kifogott mennyiségére vagy versenyzőre lebontva ki mennyit mit és mikor fogta a halat. Persze látható az adatbázisba, hogy kik vettek részt és kik voltak a versenybírók. A hal típus jelöli, hogy milyen típusú a hal például keszeg ez egy egyed. Vannak neki tulajdonságai típusa és az indexe ezenfelül, hogy őshonos-e és persze ragadozó vagy növényevő-e. Az index elsődleges A kifogott hal egyed tartalmazza a kifogott halat ki mikor és mekkorát fogott. A Versenyző egyed tartalmazza a versenyző adatait, amihez kapcsolódik a díj egyed, ami tartalmazza azokat a tulajdonságokat, hogy melyik versenyző milyen díjakat nyert el. A helyszín egyed, ami kapcsolódik a versenyzőhöz tartalmazza, hogy melyik versenyző melyik horgász helyen ült. A verseny bíró egyed tartalmazza, hogy melyik versenyzőnél volt jelen az adott bíró, és a bírónak is el van tárolva a neve meg a születési adatok.

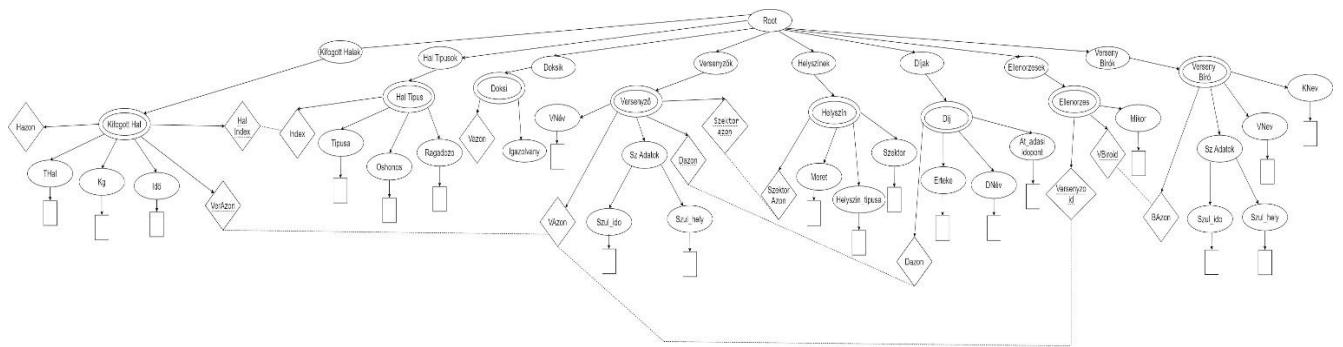
1.Feladat

1a) Az adatbázis ER modell



1b) Az adatbázis ER modell

ER modellt XDM modellre úgy konvertálhatunk, hogy három féle jelölést használunk. Van az elem, amely az XDM modellben ellipszissel lehet ábrázolni. Az attribútumot rombuszsal lehet jelölni, illetve a szöveget a téglalapban. Ezeken felül van még a dupla ellipszis, amely a többszörös előfordulási lehetőséget határozza meg. Ezenkívül minden egyedhez és N:M kapcsolathoz felvesszük az egyed többszám alakját is.



1c) Az XDM modell alapján XML dokumentum készítése

```
<?xml version="1.0" encoding="UTF-8"?>
<root xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="XMLSchemaD1HKJZ.xsd">
  <Versenyzok>
    <Versenyzo VAzon="1" Szektorazon="1" Dazon="1">
      <VNev>Bence</VNev>
      <Sz_adatok>
        <Szul_ido>1997.01.06</Szul_ido>
        <Szul_hely>Gyula</Szul_hely>
      </Sz_adatok>
    </Versenyzo>
    <Versenyzo VAzon="2" Szektorazon="2" Dazon="2">
      <VNev>Akos</VNev>
      <Sz_adatok>
        <Szul_ido>2000.10.15</Szul_ido>
        <Szul_hely>Miskolc</Szul_hely>
      </Sz_adatok>
    </Versenyzo>
    <Versenyzo VAzon="3" Szektorazon="3" Dazon="3">
      <VNev>Alex</VNev>
      <Sz_adatok>
        <Szul_ido>1999.03.20</Szul_ido>
        <Szul_hely>Budapest</Szul_hely>
      </Sz_adatok>
    </Versenyzo>
  </Versenyzok>
  <Doksik>
    <Doksi Vazon="1">
      <Igazolvany>Taj_Kartya</Igazolvany>
    </Doksi>
    <Doksi Vazon="2">
      <Igazolvany>Vezetoi_Engedely</Igazolvany>
    </Doksi>
    <Doksi Vazon="3">
      <Igazolvany>Szig_szam</Igazolvany>
    </Doksi>
    <Doksi Vazon="4">
      <Igazolvany>Lakcim_kartya</Igazolvany>
    </Doksi>
  </Doksik>
  <Verseny_Biro>
    <Verseny_Biro BAzon="1">
      <VNev>Kiss</VNev>
      <KNev>Joe</KNev>
      <Sz_adatok>
        <Szul_ido>1980.06.20</Szul_ido>
        <Szul_hely>Budapest</Szul_hely>
      </Sz_adatok>
    </Verseny_Biro>
    <Verseny_Biro BAzon="2">
      <VNev>Nagy</VNev>
      <KNev>Ferenc</KNev>
      <Sz_adatok>
        <Szul_ido>1985.12.03</Szul_ido>
        <Szul_hely>Sopron</Szul_hely>
      </Sz_adatok>
    </Verseny_Biro>
    <Verseny_Biro BAzon="3">
      <VNev>Kovacs</VNev>
    </Verseny_Biro>
  </Verseny_Biro>
</root>
```

```

        <KNev>Janos</KNev>
        <Sz_adatok>
            <Szul_ido>1975.02.10</Szul_ido>
            <Szul_hely>Gyor</Szul_hely>
        </Sz_adatok>
    </Verseny_Biro>
</Verseny_Birok>
<Kifogott_Halak>
    <Kifogott_hal Hazon="1" Halindex="1" VerAzon="1">
        <THal>Csuka</THal>
        <Kg>5</Kg>
        <ido>14:00</ido>
    </Kifogott_hal>
    <Kifogott_hal Hazon="2" Halindex="2" VerAzon="2">
        <THal>Ponty</THal>
        <Kg>10</Kg>
        <ido>15:00</ido>
    </Kifogott_hal>
    <Kifogott_hal Hazon="3" Halindex="3" VerAzon="3">
        <THal>Amur</THal>
        <Kg>8</Kg>
        <ido>17:00</ido>
    </Kifogott_hal>
</Kifogott_Halak>
<Ellenorzesek>
    <Ellenorzes VersenyzoId="2" VBiroId="1">
        <Mikor>17:55</Mikor>
    </Ellenorzes>
    <Ellenorzes VersenyzoId="1" VBiroId="1">
        <Mikor>17:50</Mikor>
    </Ellenorzes>
    <Ellenorzes VersenyzoId="3" VBiroId="3">
        <Mikor>17:45</Mikor>
    </Ellenorzes>
    <Ellenorzes VersenyzoId="2" VBiroId="3">
        <Mikor>17:40</Mikor>
    </Ellenorzes>
    <Ellenorzes VersenyzoId="1" VBiroId="2">
        <Mikor>17:35</Mikor>
    </Ellenorzes>
</Ellenorzesek>
<Dijak>
    <Dij Dazon="1">
        <DNev>Legtobb_Hal</DNev>
        <Erteke>5000</Erteke>
        <At_adasi_idopont>18:00</At_adasi_idopont>
    </Dij>
    <Dij Dazon="2">
        <DNev>Legnagyobb_Hal</DNev>
        <Erteke>5000</Erteke>
        <At_adasi_idopont>18:30</At_adasi_idopont>
    </Dij>
    <Dij Dazon="3">
        <DNev>Gyoztes</DNev>
        <Erteke>20000</Erteke>
        <At_adasi_idopont>19:00</At_adasi_idopont>
    </Dij>
</Dijak>
<Helyszinek>
    <Helyszin SzektorAzon="1">
        <Helyszin_tipusa>Alacsony_part</Helyszin_tipusa>
    </Helyszin>
</Helyszinek>

```

```
<Meret>2033</Meret>
<Szektor>A</Szektor>
</Helyszin>
<Helyszin SzektorAzon="2">
  <Helyszin_tipusa>Magas_part</Helyszin_tipusa>
  <Meret>25</Meret>
  <Szektor>B</Szektor>
</Helyszin>
<Helyszin SzektorAzon="3">
  <Helyszin_tipusa>Alacsony_part</Helyszin_tipusa>
  <Meret>30</Meret>
  <Szektor>C</Szektor>
</Helyszin>
</Helyszinek>
<Hal_Tipusok>
  <Hal_Tipus index="1">
    <Ragadozo>igen</Ragadozo>
    <Tipusa>VedettHal</Tipusa>
    <Oshonos>nem</Oshonos>
  </Hal_Tipus>
  <Hal_Tipus index="2">
    <Ragadozo>nem</Ragadozo>
    <Tipusa>VedettHal</Tipusa>
    <Oshonos>igen</Oshonos>
  </Hal_Tipus>
  <Hal_Tipus index="3">
    <Ragadozo>igen</Ragadozo>
    <Tipusa>NemVedettHal</Tipusa>
    <Oshonos>igen</Oshonos>
  </Hal_Tipus>
</Hal_Tipusok>
</root>
```

1d) Az XML dokumentum alapján XMLSchema készítése

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="root">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Versenyzo">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Versenyzo" maxOccurs="1000">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="VNev"
type="xs:string" />
                    <xs:element name="Sz_adatok">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element
name="Szul_ido" type="szulido" />
                          <xs:element
name="Szul_hely" type="xs:string" />
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                  <xs:attribute name="VAzon"
type="xs:string" />
                  <xs:attribute name="Szektorazon"
type="xs:string" />
                  <xs:attribute name="Dazon"
type="xs:string" />
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="Doksik">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Doksi" maxOccurs="1000">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Igazolvany"
type="xs:string" />
                  </xs:sequence>
                  <xs:attribute name="Vazon"
type="xs:string" />
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="Verseny_Birok">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Verseny_Biro" minOccurs="1"
maxOccurs="1000">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="VNev"
```



```

type="xs:string" />
type="xs:string" />
name="Szul_ido" type="szulido" />
name="Szul_hely" type="xs:string" />
type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="BAzon"
maxOccurs="1000">
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Kifogott_Halak">
<xs:complexType>
<xs:sequence>
<xs:element name="Kifogott_hal" minOccurs="1"
type="xs:string"/>
type="xs:string"/>
type="xs:string"/>
type="xs:string"/>
type="xs:string"/>
type="xs:string"/>
type="xs:string"/>
type="xs:string"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Ellenorzesek">
<xs:complexType>
<xs:sequence>
<xs:element name="Ellenorzes" minOccurs="1"
maxOccurs="1000">
<xs:complexType>
<xs:sequence>
<xs:element name="Mikor"
type="idoType"/>
type="xs:string"/>
type="xs:string"/>
</xs:sequence>
<xs:attribute name="VersenyzoId"
<xs:attribute name="VBiroId"
</xs:complexType>

```

```

        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="DiJak">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Dij" minOccurs="1"
maxOccurs="1000">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="DNev"
type="xs:string"/>
              <xs:element name="Erteke"
type="xs:string"/>
              <xs:element name="At_adasi_idopont"
type="idoType"/>
            </xs:sequence>
            <xs:attribute name="Dazon"
type="xs:string"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Helyszinek">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Helyszin" minOccurs="1"
maxOccurs="1000">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Helyszin_tipusa"
type="xs:string"/>
              <xs:element name="Meret"
type="xs:positiveInteger"/>
              <xs:element name="Szektor"
type="xs:string"/>
            </xs:sequence>
            <xs:attribute name="SzektorAzon"
type="xs:string"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Hal_Tipusok">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Hal_Tipus" minOccurs="1"
maxOccurs="1000">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Ragadozo"
type="xs:string"/>
              <xs:element name="Tipusa"
type="xs:string"/>
              <xs:element name="Oshonos"
type="xs:string"/>
            </xs:sequence>
            <xs:attribute name="index"

```

```

type="xs:string"/>
    </xs:complexType>
  </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
<xs:key name="di_j_Key">
  <xs:selector xpath="Dij"/>
  <xs:field xpath="@Dazon"/>
</xs:key>
<xs:key name="Versenyzo_Key">
  <xs:selector xpath="Versenyzo"/>
  <xs:field xpath="@VAzon"/>
</xs:key>
<xs:key name="kifogottHal_Key">
  <xs:selector xpath="Kifogott_hal"/>
  <xs:field xpath="@Hazon"/>
</xs:key>
<xs:key name="VersenyBir_Key">
  <xs:selector xpath="Verseny_Biro"/>
  <xs:field xpath="@Bazon"/>
</xs:key>
<xs:key name="Hal_Tipus_key">
  <xs:selector xpath="Hal_Tipus"/>
  <xs:field xpath="@index"/>
</xs:key>
<xs:key name="helyszin_key">
  <xs:selector xpath="Helyszin"/>
  <xs:field xpath="@SzektorAzon"/>
</xs:key>
<xs:keyref name="Fkey_dij_Versenyzo" refer="di_j_Key">
  <xs:selector xpath="Versenyzo"/>
  <xs:field xpath="@Dazon"/>
</xs:keyref>
<xs:keyref name="Fkey_H-V_VersenyBiro" refer="VersenyBir_Key">
  <xs:selector xpath="Ellenorzes"/>
  <xs:field xpath="@VBiroId"/>
</xs:keyref>
<xs:keyref name="Fkey_H-V_Versenyzo" refer="Versenyzo_Key">
  <xs:selector xpath="Ellenorzes"/>
  <xs:field xpath="@VersenyzoId"/>
</xs:keyref>
<xs:keyref name="Fkey_Hal_Tipus_Kifogott_hal"
refer="Hal_Tipus_key">
  <xs:selector xpath="Kifogott_hal"/>
  <xs:field xpath="@Halindex"/>
</xs:keyref>
<xs:keyref name="Fkey_Kifogott_hal_Versenyzo"
refer="Versenyzo_Key">
  <xs:selector xpath="Kifogott_hal"/>
  <xs:field xpath="@VerAzon"/>
</xs:keyref>
<xs:keyref name="Fkey_Helyszin_Versenyzo" refer="helyszin_key">
  <xs:selector xpath="Versenyzo"/>
  <xs:field xpath="@Szektorazon"/>
</xs:keyref>
<xs:keyref name="Fkey_Doksi_Versenyzo" refer="Versenyzo_Key">
  <xs:selector xpath="Doksi"/>
  <xs:field xpath="@Vazon"/>

```

```

        </xs:keyref>
    </xs:element>
    <xs:simpleType name="idoType">
        <xs:restriction base="xs:string">
            <xs:pattern value="[0-9]{1}[0-9]{1}:[0-9]{1}[0-9]{1}" />
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="szulido">
        <xs:restriction base="xs:string">
            <xs:pattern value="\d{4}[.]\d{2}[.]\d{2}" />
            <xs:length value="10" />
        </xs:restriction>
    </xs:simpleType>
</xs:schema>

```

2. feladat

2a) adatolvasás

Dom segítségével ki listázza az összes adatott.

```

package hu.dompars.Dlhkjz;

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

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;
import org.xml.sax.SAXException;

public class DomReadDlhkjz {
    public static void main(String[] args){
        NodeList list;
        try {
            DocumentBuilderFactory
factory=DocumentBuilderFactory.newInstance();
            DocumentBuilder documentBuilder=factory.newDocumentBuilder();
            //fájl beolvasása
            Document document=documentBuilder.parse(new File("XMLDlhkjz.xml"));
            document.getDocumentElement().normalize();
            //Gyökér elem megkeresése
            System.out.println("Root element : " +
document.getDocumentElement().getNodeName());
            System.out.println("-----");
            //Aktuális elem meghatározása
            list=document.getElementsByTagName("Versenyzo");

            for (int i=0;i<list.getLength();i++) {
                Node node=list.item(i);
                System.out.println("\nAktuális elem: " +
node.getNodeName());
                //Versenyzok adatainak kiírása
                if (node.getNodeType() == Node.ELEMENT_NODE) {

```

```

        Element element=(Element) node;
        System.out.println("Versenyző id: " +
element.getAttribute("VAzon"));
        System.out.println("Szektorazonosító id: " +
element.getAttribute("Szektorazon"));
        System.out.println("Díj id: " +
element.getAttribute("DAzon"));
        System.out.println("VersenyzőNév: "
+
element.getElementsByTagName("VNev").item(0).getTextContent());

        //Versenyző születési adatainak kiírása
        Node nodeszuladat=list.item(i);
        if(nodeszuladat.getNodeType()==Node.ELEMENT_NODE){
            Element elementszuldat=(Element) node;
            System.out.println("Születési idő: " +
elementszuldat.getElementsByTagName("Szul_ido").item(0).getTextContent());
            System.out.println("Születési hely: " +
elementszuldat.getElementsByTagName("Szul_hely").item(0).getTextContent());
        }
    }
    //Aktuális elem meghatározása
    list=document.getElementsByTagName("Doksi");

    for(int i=0;i<list.getLength();i++){
        Node node=list.item(i);
        System.out.println("\nAktuális elem: " + node.getNodeName());
        //Doksi adatainak kiírása
        if(node.getNodeType() == Node.ELEMENT_NODE) {
            Element element=(Element) node;
            System.out.println("Dokumentum id: " +
element.getAttribute("VAzon"));
            System.out.println("Igazolvány: " +
element.getElementsByTagName("Igazolvány").item(0).getTextContent());
        }
    }
    //Aktuális elem meghatározása
    list=document.getElementsByTagName("Verseny_Biro");

    for(int i=0;i<list.getLength();i++){
        Node node=list.item(i);
        System.out.println("\nAktuális elem: " + node.getNodeName());
        //Verseny Bírók adatainak kiírása
        if(node.getNodeType() == Node.ELEMENT_NODE) {
            Element element=(Element) node;
            System.out.println("Verseny Bíró id: " +
element.getAttribute("BAzon"));
            System.out.println("Vezetéknév: " +
element.getElementsByTagName("VNev").item(0).getTextContent());
            System.out.println("Keresztnév: " +
element.getElementsByTagName("KNev").item(0).getTextContent());

            //Bíró születési adatainak kiírása
            Node nodeszuladat=list.item(i);
            if(nodeszuladat.getNodeType()==Node.ELEMENT_NODE){
                Element elementszuldat=(Element) node;
                System.out.println("Születési idő: " +
elementszuldat.getElementsByTagName("Szul_ido").item(0).getTextContent());
                System.out.println("Születési hely: " +
elementszuldat.getElementsByTagName("Szul_hely").item(0).getTextContent());
            }
        }
    }

```

```

    }
}

//Aktuális elem meghatározása
list=document.getElementsByTagName("Kifogott_hal");

for(int i=0;i<list.getLength();i++){
    Node node=list.item(i);
    System.out.println("\nAktuális elem: " + node.getNodeName());
    //Doksi adatainak kiírása
    if(node.getNodeType() == Node.ELEMENT_NODE) {
        Element element=(Element) node;
        System.out.println("Hal id: " +
element.getAttribute("Vazon"));
        System.out.println("KifogottHalnak az idje: " +
element.getAttribute("Halindex"));
        System.out.println("Versenyző idje: " +
element.getAttribute("VerAzon"));
        System.out.println("Tipusa: " +
element.getElementsByTagName("THal").item(0).getTextContent());
        System.out.println("Mérete: " +
element.getElementsByTagName("Kg").item(0).getTextContent());
        System.out.println("Mikor fogták: " +
element.getElementsByTagName("ido").item(0).getTextContent());

    }
}

//Aktuális elem meghatározása
list=document.getElementsByTagName("Ellenorzes");

for(int i=0;i<list.getLength();i++){
    Node node=list.item(i);
    System.out.println("\nAktuális elem: " + node.getNodeName());
    //Doksi adatainak kiírása
    if(node.getNodeType() == Node.ELEMENT_NODE) {
        Element element=(Element) node;
        System.out.println("Biro id: " +
element.getAttribute("VBiroId"));
        System.out.println("Versenyző idje: " +
element.getAttribute("VersenyzoId"));
        System.out.println("Mikor: " +
element.getElementsByTagName("Mikor").item(0).getTextContent());
    }
}

//Aktuális elem meghatározása
list=document.getElementsByTagName("Dij");

for(int i=0;i<list.getLength();i++){
    Node node=list.item(i);
    System.out.println("\nAktuális elem: " + node.getNodeName());
    //Doksi adatainak kiírása
    if(node.getNodeType() == Node.ELEMENT_NODE) {
        Element element=(Element) node;
        System.out.println("Dij id: " +
element.getAttribute("VBiroId"));
        System.out.println("Értéke: " +
element.getElementsByTagName("Erteke").item(0).getTextContent());
        System.out.println("Át adási időpont: " +
element.getElementsByTagName("At_adasi_idopont").item(0).getTextContent());
    }
}

```

```

//Aktuális elem meghatározása
list=document.getElementsByTagName("Helyszin");

for(int i=0;i<list.getLength();i++){
    Node node=list.item(i);
    System.out.println("\nAktuális elem: " + node.getNodeName());
    //Doksi adatainak kiírása
    if(node.getNodeType() == Node.ELEMENT_NODE) {
        Element element=(Element) node;
        System.out.println("Szektor id: " +
element.getAttribute("SzektorAzon"));
        System.out.println("Helyszin tipusa: " +
element.getElementsByTagName("Helyszin_tipusa").item(0).getTextContent());
        System.out.println("Mérete: " +
element.getElementsByTagName("Meret").item(0).getTextContent());
        System.out.println("Szektor: " +
element.getElementsByTagName("Szektor").item(0).getTextContent());
    }
}

//Aktuális elem meghatározása
list=document.getElementsByTagName("Hal_Tipus");

for(int i=0;i<list.getLength();i++){
    Node node=list.item(i);
    System.out.println("\nAktuális elem: " + node.getNodeName());
    //Doksi adatainak kiírása
    if(node.getNodeType() == Node.ELEMENT_NODE) {
        Element element=(Element) node;
        System.out.println("Hal_tipus id: " +
element.getAttribute("index"));
        System.out.println("Ragadozó: " +
element.getElementsByTagName("Ragadozo").item(0).getTextContent());
        System.out.println("Tipusa: " +
element.getElementsByTagName("Tipusa").item(0).getTextContent());
        System.out.println("Oshonos: " +
element.getElementsByTagName("Oshonos").item(0).getTextContent());
    }
}

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

```

2b) adatmódosítás

A hal típus táblából kiírja ki listázza azokat amelyek védetek, és kiírja a consolera.

```

package hu.dompars.Dlhkjz;

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;

```

```

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import java.io.File;
import java.io.IOException;

public class DOMQueryDlhcjz {
    public static void main(String[] args) {
        try {
            NodeList nodeList;

            DocumentBuilderFactory factory
=DocumentBuilderFactory.newInstance();
            DocumentBuilder builder=factory.newDocumentBuilder();
            //Fájl beolvasása
            Document document=builder.parse(new File("XMLDlhcjz.xml"));
            document.getDocumentElement().normalize();

            //Aktuális elem meghatározása
            nodeList=document.getElementsByTagName("Hal_Tipus");

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

                //Hal típusok adatainak kiírása
                if (node.getNodeType()==Node.ELEMENT_NODE) {
                    Element element=(Element) node;
                    String tipus=
element.getElementsByTagName("Tipusa").item(0).getTextContent();

                    if (tipus.equals("VedettHal"))
                    {
                        System.out.println("\nAktuális elem: " +
node.getNodeName());

                        System.out.println("-----");
                        System.out.println("");
                        System.out.println("Hal típus id: " +
element.getAttribute("index"));
                        System.out.println("Ragadozó: " +
element.getElementsByTagName("Ragadozo").item(0).getTextContent());
                        System.out.println("Tipusa: " +
element.getElementsByTagName("Tipusa").item(0).getTextContent());
                        System.out.println("Oshonos: " +
element.getElementsByTagName("Oshonos").item(0).getTextContent());

                    }
                }
            }
        } catch (ParserConfigurationException | IOException | SAXException
e) {
            e.printStackTrace();
        }
    }
}

```

3c) adatlekérdezés

```

package hu.domparse.Dlhcjz;

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;

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

public class DOMQueryDlhkjz {
    public static void main(String[] args) {
        try {
            DocumentBuilderFactory factory =
DocumentBuilderFactory.newInstance();
            DocumentBuilder builder = factory.newDocumentBuilder();
            // fájl beolvasása
            Document document = builder.parse(new File("XMLDlhkjz.xml"));
            document.getDocumentElement().normalize();
            //Xpath készítése
            XPath xPath= XPathFactory.newInstance().newXPath();

            //meg kell adni az elérési út kifejezést és a csomópont listát

            //Összes versenyző lekérdezése
            //String expression="root/Versenyzok/Versenyzo";
            //az utolso versenyzo lekérdezése
            String expression="root/Versenyzok/Versenyzo[last()]";

            //Készítsünk egy listát, majd a Path kifejezést meg kell írni
és ki kell értékelni.
            NodeList nodeList=(NodeList)
xPath.compile(expression).evaluate(document, XPathConstants.NODESET);

            //A for ciklus segítségével a NodeList csomópontjait végig kell
iterrálni.
            for (int i=0;i<nodeList.getLength();i++){
                Node node=nodeList.item(i);
                System.out.println ("\naktuális elem: " +
node.getNodeName());

                //meg kell vizsgálni a cssomópontot, az subelement
tesztelése megtörtént
                if(node.getNodeType() == Node.ELEMENT_NODE &&
node.getNodeName().equals("Versenyzo")){
                    Element element=(Element) node;
                    System.out.println("Versenyzo id: " +
element.getAttribute("VAzon"));
                    System.out.println("Szektorazonosító id: " +
element.getAttribute("Szektorazon"));
                    System.out.println("Díj id: " +
element.getAttribute("Dazon"));
                    System.out.println("VersenyzőNév: "
+
element.getElementsByTagName("VNev").item(0).getTextContent());
                }
            }
        }
    }
}

```

```
        }  
    } catch (ParserConfigurationException | XPathExpressionException |  
IOException | SAXException e){  
        e.printStackTrace();  
    }  
}  
}
```