

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Vadásztársaság

Készítette: Jordán Kornél

Neptunkód: DWWAP

Dátum: 2022.11.10.

A feladatban egy vadászvizsga lebonyolítását biztosító adatbázis modellt fogok bemutatni.

Az egyedei: Egyesület, Felügyelő, Vadász, Vizsga

Egyesület egyed:

Az Egyesület egyed az adott vadászvizsga lebonyolítását biztosító egyesületnek az adatait tartalmazza. Van benne: E_ID, ami kulcs típusú, E_Név, alapítás_Éve, Telekhely.

Felügyelő egyed:

A felügyelő egyed azokat a személyeket mutatja be, akik felügyelik az adott vizsgázót. Van benne: Személyi_IG, ami kulcs típusú, Név ebben van Vezetéknév és Keresztnév, Telefonszám, Email_cím

Vadász egyed:

A vadász egyedben az adott vizsgázó személyek adatai vannak benne. Van benne: Személyi_IG ami kulcs típusú, Név ebben van Vezetéknév és Keresztnév, Kor, Telefonszám, Email_cím, Elért_pontszám.

Vizsga egyed:

A vizsga egyed az adott vizsga adatait tárolja. Van benne: Vizsga_ID ami kulcs típusú, van Elmélet amiben van Írásbeli és Szóbeli rész. Van még Gyakorlat, Helyszín, amiben van Irányító_szám, Település, amiben van Utca és Házszám.

Kapcsolatok:

Vizsga – Egyesület: 1:N kapcsolat, mivel egy vizsga csak egy egyesülethez tartozhat, de egy egyesületnek lehet több vizsgája is. (Egyesület-vizsga)

Vadász – Egyesület: 1:1 kapcsolat, mivel egy egyesületnek csak egy elnöke lehet (Elnök)

Vadász – Felügyelő: N:M kapcsolat, mivel egy vadásznak lehet több felügyelője is és egy felügyelő felügyelni tud több vadászt is. (Vadász-felügyelő)

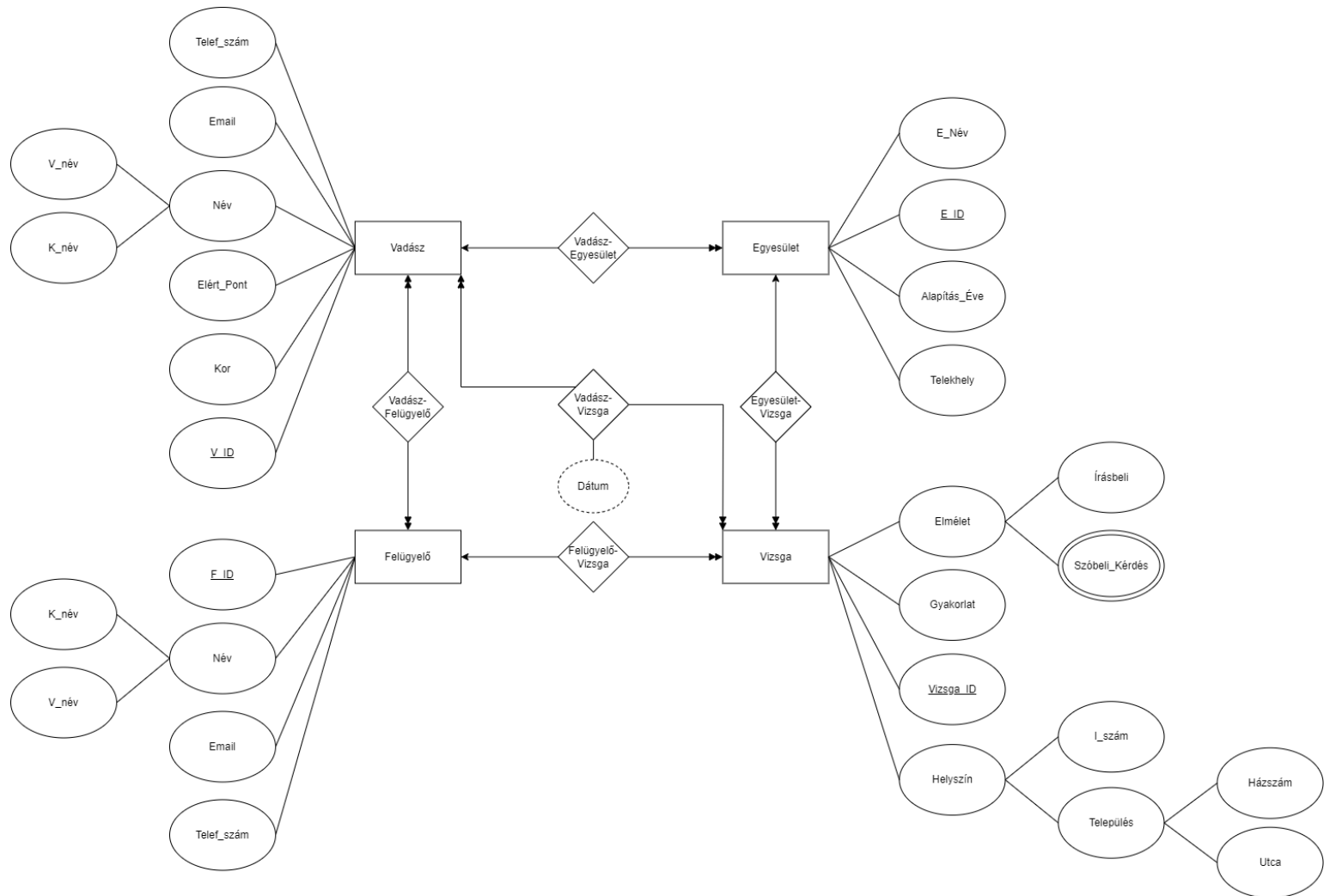
Vadász – Vizsga: N:M kapcsolat, mivel több vadász jelentkezhet több vizsgára. (Vadász- vizsga)

Vizsga – Felügyelő: 1:N kapcsolat, mivel egy vizsga az több felügyelőhöz tartozik, de egy felügyelő csak egy vizsgán tud egyszerre részt venni. (felügyelő-vizsga)

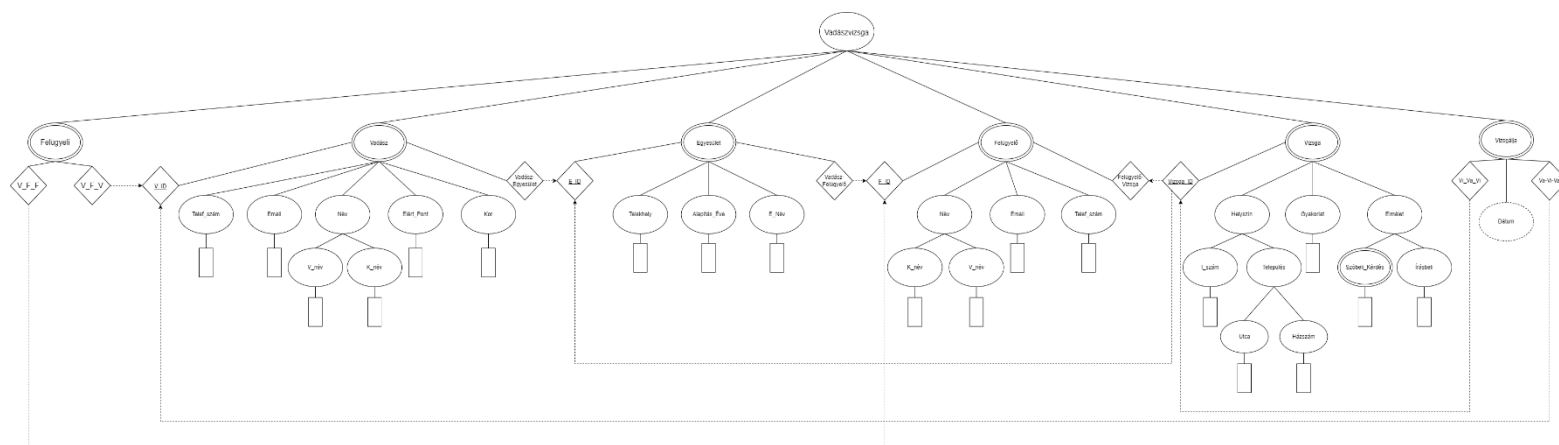
A Szóbeli_Kérdés egy többértékű tulajdonság, mivel annak van Sorszám és Szöveg része is.

1a) Az adatbázis ER modell: A modell egy vadásztársaságon belüli vizsgát mutat be

1.



1b) Az adatbázis konvertálása XDM modellre: Az 1a) feladatban megadott XDM modell konvertálása szabvány alapján



1c) Az XDM modell alapján XML dokumentum készítése: Az 1b) feladat ábrájának átírása XML dokumentumra

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

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

```
<vadasz vadasz-egyesulet="ve1" V_ID="1">
  <telef_szam>0620222222</telef_szam>
  <Email>valami@Email.com</Email>
  <V_nev>Kiss</V_nev>
  <K_nev>Pista</K_nev>
  <elert_pont>5</elert_pont>
  <kor>30</kor>
</vadasz>
```

```
<vadasz vadasz-egyesulet="ve2" V_ID="2">
  <telef_szam>0620222223</telef_szam>
  <Email>valami2@Email.com</Email>
  <V_nev>Nagy</V_nev>
  <K_nev>Pista</K_nev>
  <elert_pont>6</elert_pont>
  <kor>40</kor>
</vadasz>
```

```
<vadasz vadasz-egyesulet="ve3" V_ID="3">
  <telef_szam>0620222224</telef_szam>
  <Email>valam3@Email.com</Email>
  <V_nev>Valamilyen</V_nev>
  <K_nev>Pista</K_nev>
```

<elert_pont>7</elert_pont>
<kor>50</kor>
</vadasz>

<egyesulet E_ID="1" vadasz-felugyelo="1">
 <telekhely>Miskolc</telekhely>
 <alapitas_eve>1969</alapitas_eve>
 <e_nev>Kis Egyesület</e_nev>
</egyesulet>

<egyesulet E_ID="2" vadasz-felugyelo="2">
 <telekhely>Pest</telekhely>
 <alapitas_eve>1980</alapitas_eve>
 <e_nev>Nagy Egyesület</e_nev>
</egyesulet>

<egyesulet E_ID="3" vadasz-felugyelo="3">
 <telekhely>Debrecen</telekhely>
 <alapitas_eve>1990</alapitas_eve>
 <e_nev>Gyors Egyesület</e_nev>
</egyesulet>

<felugyelo F_ID="1" felugyelo-vizsga="1">
 <V_nev>Kiss</V_nev>
 <K_nev>Lajos</K_nev>
 <Email>lajos@gmail.com</Email>
 <telef_szam>06203334444</telef_szam>
</felugyelo>

<felugyelo F_ID="2" felugyelo-vizsga="2">
 <V_nev>Nagy</V_nev>
 <K_nev>Lajos</K_nev>
 <Email>lajosanagy@gmail.com</Email>
 <telef_szam>06203334456</telef_szam>
</felugyelo>

<felugyelo F_ID="3" felugyelo-vizsga="3">
 <V_nev>Kiss</V_nev>
 <K_nev>Ferenc</K_nev>
 <Email>ferenc@gmail.com</Email>
 <telef_szam>06203338888</telef_szam>
</felugyelo>

<vizsga vizsga_ID="1">
 <l_szam>3666</l_szam>
 <utca>Petőfi</utca>
 <hazszam>1</hazszam>
 <gyakorlat>Kinti</gyakorlat>
 <szobeli_kerdes>Valami1</szobeli_kerdes>
 <irasbeli>Valami3</irasbeli>
</vizsga>

<vizsga vizsga_ID="2">
 <l_szam>3700</l_szam>
 <utca>Nagy</utca>
 <hazszam>7</hazszam>
 <gyakorlat>Kinti</gyakorlat>
 <szobeli_kerdes>Valami7</szobeli_kerdes>
 <irasbeli>Valami4</irasbeli>
</vizsga>

<vizsga vizsga_ID="3">
 <l_szam>3800</l_szam>
 <utca>Fekete</utca>
 <hazszam>30</hazszam>
 <gyakorlat>Kinti</gyakorlat>
 <szobeli_kerdes>Valami3</szobeli_kerdes>
 <irasbeli>Valami8</irasbeli>
</vizsga>

<vizsgalja Vi_Va_Vi="1" Va_Vi_Va="1">
 <datum>2022.11.09.</datum>
</vizsgalja>

<vizsgalja Vi_Va_Vi="2" Va_Vi_Va="2">
 <datum>2022.11.16.</datum>
</vizsgalja>

<vizsgalja Vi_Va_Vi="3" Va_Vi_Va="3">
 <datum>2022.11.23.</datum>
</vizsgalja>

<felugyeli V_F_F="VFF1" V_F_V="VFV1"></felugyeli>
<felugyeli V_F_F="VFF2" V_F_V="VFV2"></felugyeli>
<felugyeli V_F_F="VFF3" V_F_V="VFV3"></felugyeli>

</vadaszvizsga>

1d) Az XML dokumentum alapján XMLSchema készítése (saját típusok, ref, key, keyref, speciális elemek): Az adatok kommentezve vannak a könnyebb olvashatóság miatt

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

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
```

```
<!-- Adatok típusai-->
```

```
<xs:element name="telef_szam" type="xs:integer" />
<xs:element name="Email" type="xs:string" />
<xs:element name="V_nev" type="xs:string" />
<xs:element name="K_nev" type="xs:string" />
<xs:element name="elert_pont" type="xs:integer" />
<xs:element name="kor" type="xs:integer" />
<xs:element name="telekhely" type="xs:string" />
<xs:element name="alapitas_eve" type="xs:integer" />
<xs:element name="e_nev" type="xs:string" />
<xs:element name="l_szam" type="xs:integer" />
<xs:element name="utca" type="xs:string" />
<xs:element name="hazszam" type="xs:integer" />
<xs:element name="gyakorlat" type="xs:string" />
<xs:element name="szobeli_kerdes" type="xs:string" />
<xs:element name="irasbeli" type="xs:string" />
<xs:element name="datum" type="xs:string" />
```

```
<xs:attribute name="V_F_F" type="xs:string" />
<xs:attribute name="V_F_V" type="xs:string" />
<xs:attribute name="V_ID" type="xs:integer" />
<xs:attribute name="vadasz-egyesulet" type="xs:string" />
<xs:attribute name="E_ID" type="xs:integer" />
<xs:attribute name="vadasz-felugyelo" type="xs:integer" />
<xs:attribute name="F_ID" type="xs:integer" />
<xs:attribute name="felugyelo-vizsga" type="xs:integer" />
<xs:attribute name="vizsga_ID" type="xs:integer" />
<xs:attribute name="Vi_Va_Vi" type="xs:integer" />
<xs:attribute name="Va_Vi_Va" type="xs:integer" />
```

```
<!-- Komplex típusok-->
```

```
<xs:complexType name="vadaszTipus">
```

```
<xs:sequence>
  <xs:element ref="telef_szam"/>
  <xs:element ref="Email"/>
  <xs:element ref="V_nev"/>
  <xs:element ref="K_nev"/>
  <xs:element ref="elert_pont"/>
  <xs:element ref="kor"/>
</xs:sequence>
<xs:attribute ref="V_ID" use="required"/>
<xs:attribute ref="vadasz-egyesulet" use="required"/>
</xs:complexType>
```

```
<xs:complexType name="egyesuletTipus">
  <xs:sequence>
    <xs:element ref="telekhely"/>
    <xs:element ref="alapitas_eve"/>
    <xs:element ref="e_nev"/>
  </xs:sequence>
  <xs:attribute ref="vadasz-felugyelo" use="required"/>
  <xs:attribute ref="E_ID" use="required"/>
</xs:complexType>
```

```
<xs:complexType name="felugyeloTipus">
  <xs:sequence>
    <xs:element ref="V_nev"/>
    <xs:element ref="K_nev"/>
    <xs:element ref="Email"/>
    <xs:element ref="telef_szam"/>
  </xs:sequence>
  <xs:attribute ref="felugyelo-vizsga" use="required"/>
  <xs:attribute ref="F_ID" use="required"/>
</xs:complexType>
```

```
<xs:complexType name="vizsgaTipus">
  <xs:sequence>
    <xs:element ref="I_szam"/>
    <xs:element ref="utca"/>
    <xs:element ref="hazszam"/>
    <xs:element ref="gyakorlat"/>
    <xs:element ref="szobeli_kerdes"/>
    <xs:element ref="irasbeli"/>
  </xs:sequence>
  <xs:attribute ref="vizsga_ID" use="required"/>
</xs:complexType>
```

```
<xs:complexType name="vizsgaljaTipus">
  <xs:sequence>
    <xs:element ref="datum"/>
  </xs:sequence>
  <xs:attribute ref="Vi_Va_Vi" use="required"/>
  <xs:attribute ref="Va_Vi_Va" use="required"/>
</xs:complexType>
<xs:complexType name="felugyeliTipus">
```



```
<xs:attribute ref="V_F_F" use="required"/>
<xs:attribute ref="V_F_V" use="required"/>
</xs:complexType>
```

```
<xs:element name="vadaszvizsga">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="vadasz" type="vadaszTipus" maxOccurs="unbounded" />
      <xs:element name="egyesulet" type="egyesuletTipus" maxOccurs="unbounded" />
      <xs:element name="felugyelo" type="felugyeloTipus" maxOccurs="unbounded" />
      <xs:element name="vizsga" type="vizsgaTipus" maxOccurs="unbounded" />
      <xs:element name="vizsgalja" type="vizsgaljaTipus" maxOccurs="unbounded" />
      <xs:element name="felugyeli" type="felugyeliTipus" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
```

```
<!-- Kulcsok -->
```

```
<xs:key name="Vadasz_kulcs">
  <xs:selector xpath="vadasz" />
  <xs:field xpath="@V_ID" />
</xs:key>
```

```
<xs:key name="egyesulet_kulcs">
  <xs:selector xpath="egyesulet" />
  <xs:field xpath="@E_ID" />
</xs:key>
```

```
<xs:key name="felugyelo_kulcs">
  <xs:selector xpath="felugyelo" />
  <xs:field xpath="@F_ID" />
</xs:key>
```

```
<xs:key name="vizsga_kulcs">
  <xs:selector xpath="vizsga" />
  <xs:field xpath="@vizsga_ID" />
</xs:key>
```

```
<!-- Idegen kulcsok-->
```

```
<xs:keyref refer="Vadasz_kulcs" name="vadasz_idegen_kulcs">
  <xs:selector xpath="egyesulet" />
  <xs:field xpath="@vadasz-egyesulet" />
</xs:keyref>
```

```
<xs:keyref refer="egyesulet_kulcs" name="vizsga_idegen_kulcs">
  <xs:selector xpath="felugyelo" />
  <xs:field xpath="@vadasz-felugyelo" />
</xs:keyref>
```

```
<xs:keyref refer="felugyelo_kulcs" name="vizsga2_idegen_kulcs">
  <xs:selector xpath="felugyeli" />
```

```
<xs:field xpath="@felugyelo-vizsga" />
</xs:keyref>
```

```
<xs:keyref refer="felugyelo_kulcs" name="vizsga3_idegen_kulcs">
  <xs:selector xpath="vizsga" />
  <xs:field xpath="@Vi_Va_Vi" />
</xs:keyref>
```

```
<xs:keyref refer="felugyelo_kulcs" name="vadasz2_idegen_kulcs">
  <xs:selector xpath="vadasz" />
  <xs:field xpath="@Va_Vi_Va" />
</xs:keyref>
```

```
<xs:keyref refer="felugyelo_kulcs" name="felugyelo_idegen_kulcs">
  <xs:selector xpath="felugyelo" />
  <xs:field xpath="@V_F_F" />
</xs:keyref>
```

```
<xs:keyref refer="vizsga_kulcs" name="vadasz3_idegen_kulcs">
  <xs:selector xpath="vadasz" />
  <xs:field xpath="@V_F_V" />
</xs:keyref>
```

```
</xs:element>
```

```
</xs:schema>
```

2. feladat

2a) adatolvasás: Egy olvasó létrehozása, bejövő adatok "formájának" megadása, azért, hogy minden adatot megfelelően olvasson be.

```
package hu.domparse.dwwuap;

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 DOMReadDWWUAP {

    public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException {
        //Forrás file
        File file = new File("XMLDWWUAP.xml");

        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = factory.newDocumentBuilder();

        Document doc = dBuilder.parse(file);

        doc.getDocumentElement().normalize();
        //Gyökérelem
        System.out.println("Root element: " + doc.getDocumentElement().getNodeName());
        //Gyerekelemek lementése
        NodeList nList = (NodeList) doc.getDocumentElement();

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

            //Ha egyesület
            if (node.getNodeName() == "egyesulet"){
                if (!node.getNodeName().equals("#text")) {
                    System.out.println("\n");
                    System.out.println("Current element: " + node.getNodeName());
                }
            }
            //Egyesület adatainak kiírása
        }
    }
}
```

```
if(node.getNodeType()==Node.ELEMENT_NODE) {
    Element elem = (Element) node;

    String e_id = elem.getAttribute("E_ID");

    Node egyesulet_node = elem.getElementsByTagName("telekhely").item(0);
    String telekhely_name = egyesulet_node.getTextContent();

    Node egyesulet_node2 = elem.getElementsByTagName("alapitas_eve").item(0);
    String alapitas_eve_name = egyesulet_node2.getTextContent();

    Node egyesulet_node3 = elem.getElementsByTagName("e_nev").item(0);
    String e_nev_name = egyesulet_node3.getTextContent();

    System.out.printf("Egyesulet id: %s%n", e_id);
    System.out.printf("Telekhely neve: %s%n", telekhely_name);
    System.out.printf("Alapitas eve: %s%n", alapitas_eve_name);
    System.out.printf("Egyesulet neve: %s%n", e_nev_name);
}
}
}
}
```

2b) adatmódosítás: Alap file beolvasása, mindig az adott kommentek alapján történő műveletek csinálása, az adattagokkal.

```
package hu.domparsedwwuap;

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

public class DOMModifyDWWUAP {

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

        try {
            //Forrás file
            File inputFile = new File("XML2DWWUAP.xml");
            DocumentBuilderFactory docFactory = DocumentBuilderFactory.newInstance();
            DocumentBuilder docBuilder = docFactory.newDocumentBuilder();
            Document doc = docBuilder.parse(inputFile);

            //Első és harmadik egyesület mentése
            Node vadasz1 = doc.getElementsByTagName("vadasz").item(0);
            Node vadasz3 = doc.getElementsByTagName("vadasz").item(2);
            //Gyökérelem
            Node vadaszvizsga = doc.getFirstChild();

            //Harmadik egyesület ID váltás
            NamedNodeMap attr = vadasz3.getAttributes();
            Node nodeAttr = attr.getNamedItem("V_ID");
            nodeAttr.setTextContent("20");

            //Második egyesület irányítószámának megváltoztatása
            NodeList list = vadasz1.getChildNodes();
            for (int i = 0; i < list.getLength(); i++) {
                Node node = list.item(i);
                if (node.getNodeType() == Node.ELEMENT_NODE) {
```

```

        Element elem = (Element) node;
        if ("telef_szam".equals(elem.getNodeName())) {
            elem.setTextContent("06206964200");
        }
    }
}

```

```

//Harmadik egyesületnek az Email megváltoztatása
NodeList list1 = vadasz3.getChildNodes();
for (int i = 0; i < list1.getLength(); i++) {
    Node node1 = list1.item(i);
    if (node1.getNodeType() == Node.ELEMENT_NODE) {
        Element elem1 = (Element) node1;
        if ("Email".equals(elem1.getNodeName())) {
            elem1.setTextContent("valamiModify@Email.com");
        }
    }
}
}

```

```

//Egyesületek törlése
NodeList childNodes = vadaszvizsga.getChildNodes();
for(int i = 0; i < childNodes.getLength(); i++) {
    Node node = childNodes.item(i);

    if("egyesulet".equals(node.getNodeName()))
        vadaszvizsga.removeChild(node);
}

```

```

//Konzolra kirás
TransformerFactory transformerFactory = TransformerFactory.newInstance();
Transformer transformer = transformerFactory.newTransformer();
DOMSource source = new DOMSource(doc);
System.out.println("New File");
StreamResult consoleResult = new StreamResult(System.out);
transformer.transform(source, consoleResult);
}catch(Exception e) {
    e.printStackTrace();
}
}
}

```

2c) adatlekérdezés: Alap beolvasó létrehozása, adatok belementése, végig megyünk az elemein kihagyva a kort.

```
package hu.domparse.dwwwuap;

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 DOMQueryDWWUAP {

    public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException {
        //Forrás file
        File file = new File("XMLDWWUAP.xml");

        DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();

        Document doc = dBuilder.parse(file);
        doc.getDocumentElement().normalize();
        //Gyökér elem
        System.out.print("Root element: ");
        System.out.println(doc.getDocumentElement().getNodeName());
        //Vadaszvizsga betöltése
        NodeList nList = doc.getElementsByTagName("vadaszvizsga");

        System.out.println("-----");

        //Végigfut a vadasznak a gyerek elemein kihagyva a kort
        for (int i = 0; i < nList.getLength(); i++) {
            Node node = nList.item(i);
            System.out.println("\nCurrent Element : "+node.getNodeName());
            if (node.getNodeType() == Node.ELEMENT_NODE) {
                Element elem = (Element) node;
                System.out.println("ID:"+elem.getAttribute("V_ID"));
                NodeList nList2 = elem.getChildNodes();
                for (int j = 0; j < nList2.getLength(); j++) {
                    Node node2 = nList2.item(j);
                    if (node2.getNodeType() == Node.ELEMENT_NODE) {
                        Element elem2 = (Element) node2;
                        if(!node2.getNodeName().equals("kor")) {
```

```
        System.out.println(node2.getNodeName()+" : "+node2.getTextContent());
    }
    NodeList nList3 = elem2.getChildNodes();
    for (int k = 0; k < nList3.getLength(); k++) {
        Node node3 = nList3.item(k);
        if (node3.getNodeType() == Node.ELEMENT_NODE) {
            System.out.println("vadaszvizsga : " + node3.getNodeName() + " : " + node3.getTextContent());
        }
    }
}
}
```