

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_száma, Település, amiben van Utca és Háza_száma.

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)

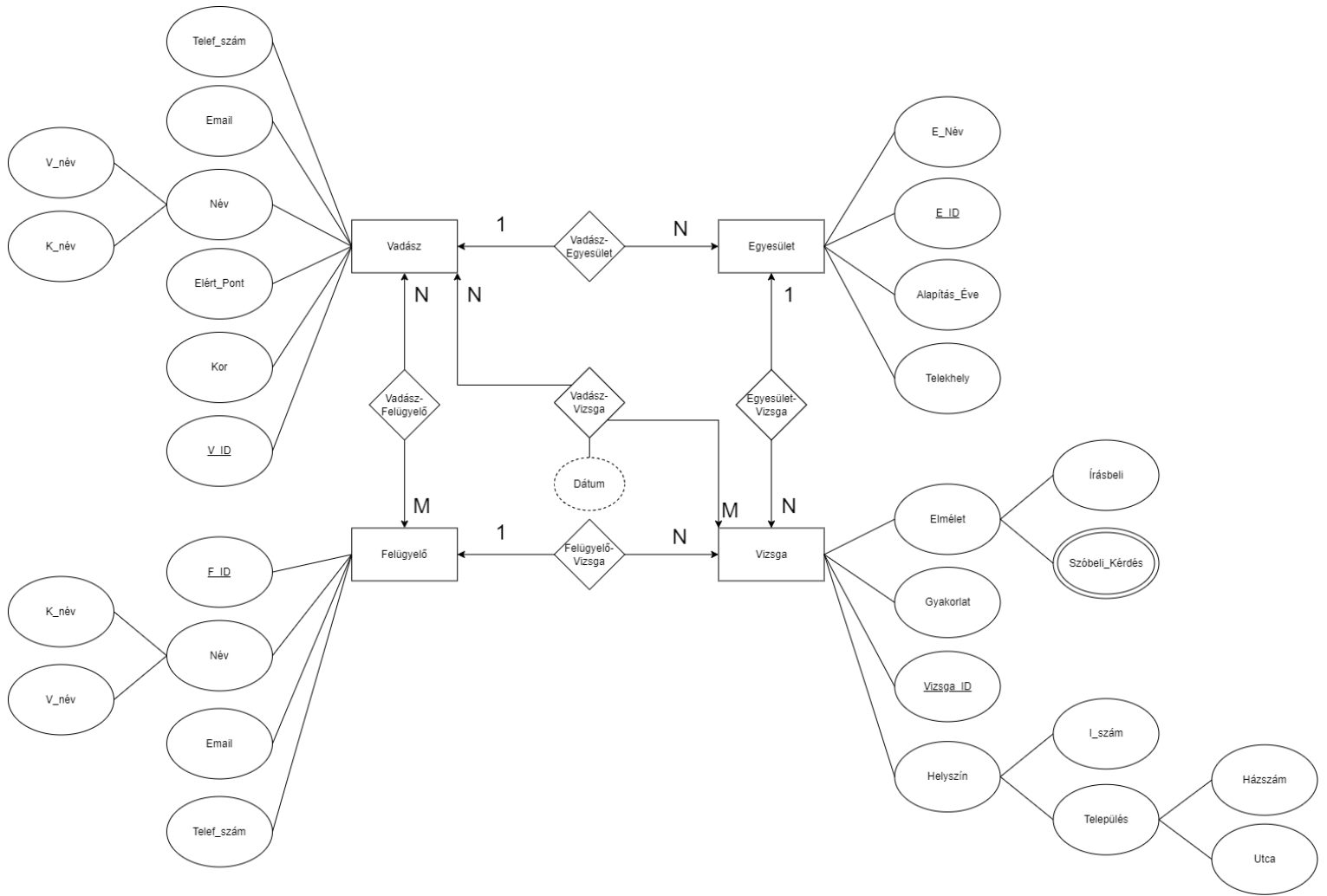
Vadász – Vizsga: N:M kapcsolat, mivel több vadász jelentkezhet több vizsgára. (Vadász-vizsga)

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

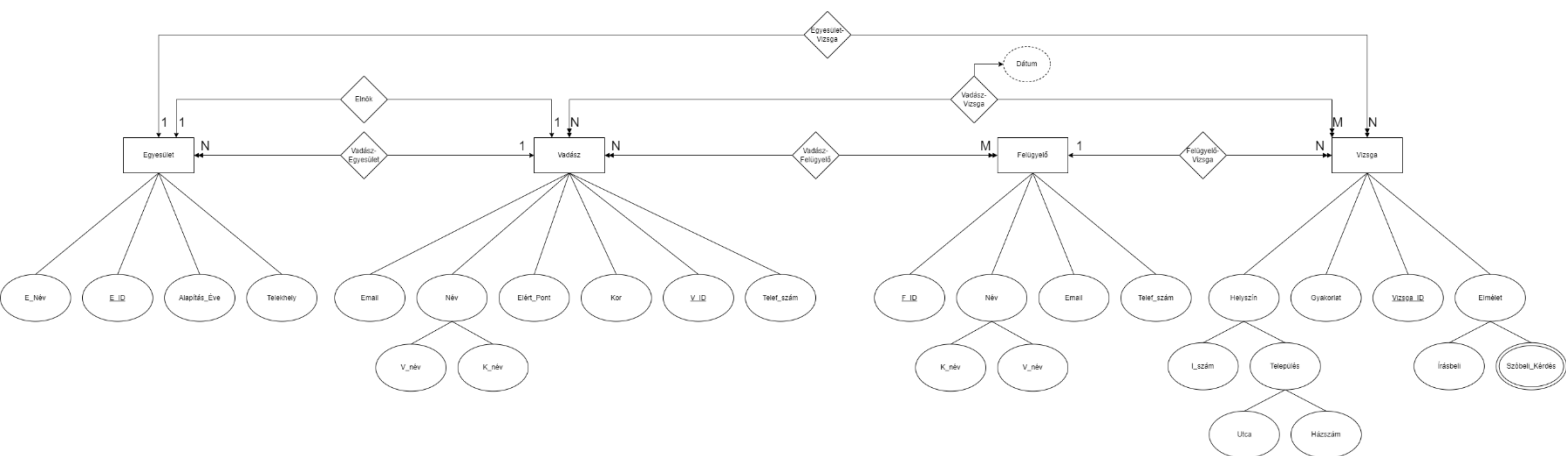
A Szóbeli_Kérdés egy többértékű tulajdonság, mivel annak van Sorszáma és Szövege is.

1a) Az adatbázis ER modell: (Két ábrát készítettem, de ugyanazt mutatják be)

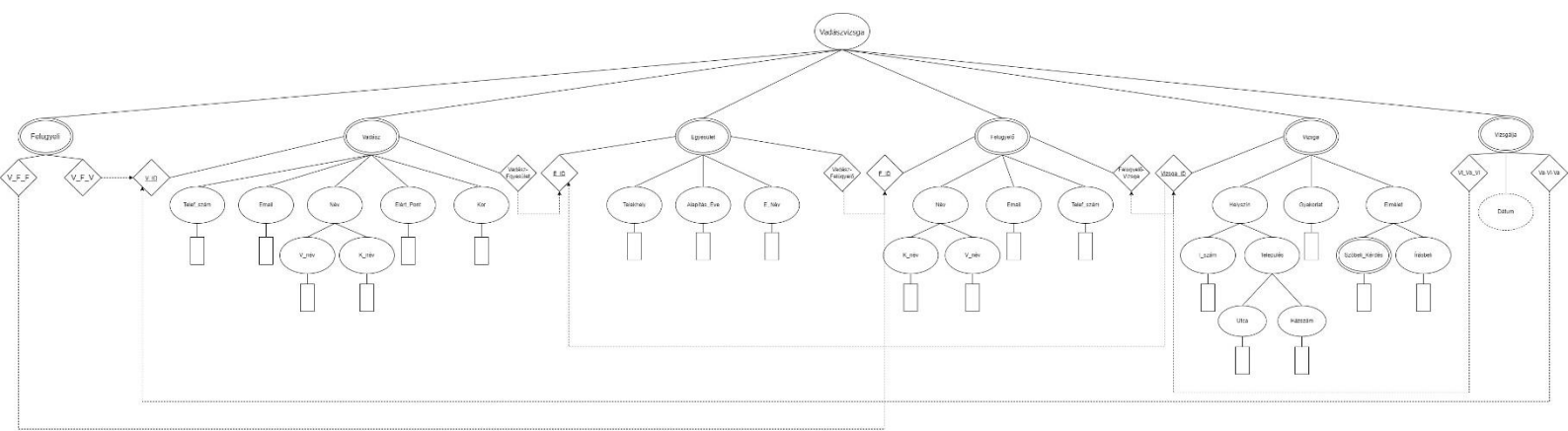
1.



2.



1b) Az adatbázis konvertálása XDM modellre



1c) Az XDM modell alapján XML dokumentum készítése:

```

XMLDWWUAP.xml X XMLSchemaDWWUAP.xsd
1. feladat > XMLDWWUAP.xml > vadaszvizsga
1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <vadaszvizsga xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="XMLSchemaDWWUAP.xsd">
4
5
6
7
8  <vadasz vadasz-egyesulet="ve1" V_ID="1">
9    <telef_szam>06202222222</telef_szam>
10   <Email>valami1@Email.com</Email>
11   <V_nev>Kiss</V_nev>
12   <K_nev>Pista</K_nev>
13   <elert_pont>5</elert_pont>
14   <kor>30</kor>
15 </vadasz>
16
17 <vadasz vadasz-egyesulet="ve2" V_ID="2">
18   <telef_szam>06202222223</telef_szam>
19   <Email>valami2@Email.com</Email>
20   <V_nev>Nagy</V_nev>
21   <K_nev>Pista</K_nev>
22   <elert_pont>6</elert_pont>
23   <kor>40</kor>
24 </vadasz>
25
26 <vadasz vadasz-egyesulet="ve3" V_ID="3">
27   <telef_szam>06202222224</telef_szam>
28   <Email>valam3@Email.com</Email>
29   <V_nev>Valamilyen</V_nev>
30   <K_nev>Pista</K_nev>
31   <elert_pont>7</elert_pont>
32   <kor>50</kor>
33 </vadasz>

```

```
XMLDWWUAP.xml X XMLSchemaDWWUAP.xsd XMLDWWUAP.xml X
1. feladat > XMLDWWUAP.xml > vadaszvizsga
37
38 <egyesulet E_ID="1" vadasz-felugyelo="1">
39   <telekhely>Miskolc</telekhely>
40   <alapitas_eve>1969</alapitas_eve>
41   <e_nev>Kis Egyesület</e_nev>
42 </egyesulet>
43
44 <egyesulet E_ID="2" vadasz-felugyelo="2">
45   <telekhely>Pest</telekhely>
46   <alapitas_eve>1980</alapitas_eve>
47   <e_nev>Nagy Egyesület</e_nev>
48 </egyesulet>
49
50 <egyesulet E_ID="3" vadasz-felugyelo="3">
51   <telekhely>Debrecen</telekhely>
52   <alapitas_eve>1990</alapitas_eve>
53   <e_nev>Gyors Egyesület</e_nev>
54 </egyesulet>
55
56
57
58
59 <felugyelo F_ID="1" felugyelo-vizsga="1">
60   <V_nev>Kiss</V_nev>
61   <K_nev>Lajos</K_nev>
62   <Email>lajos@gmail.com</Email>
63   <telef_szam>0620333444</telef_szam>
64 </felugyelo>
65
66 <felugyelo F_ID="2" felugyelo-vizsga="2">
67   <V_nev>Nagy</V_nev>
68   <K_nev>Lajos</K_nev>
69   <Email>lajosnagy@gmail.com</Email>
70   <telef_szam>06203334456</telef_szam>
71 </felugyelo>
72
73 <felugyelo F_ID="3" felugyelo-vizsga="3">
74   <V_nev>Kiss</V_nev>
75   <K_nev>Ferenc</K_nev>
76   <Email>ferenc@gmail.com</Email>
77   <telef_szam>0620333888</telef_szam>
78
79
80
81
82
83 <vizsga vizsga ID="1">
84   <I_szam>3666</I_szam>
85   <utca>Petőfi</utca>
86   <hazszam>1</hazszam>
87   <gyakorlat>Kinti</gyakorlat>
88   <szobeli_kerdes>Valami1</szobeli_kerdes>
89   <irasbeli_kerdes>Valami3</irasbeli_kerdes>
90 </vizsga>
91
92 <vizsga vizsga ID="2">
93   <I_szam>3700</I_szam>
94   <utca>Nagy</utca>
95   <hazszam>7</hazszam>
96   <gyakorlat>Kinti</gyakorlat>
97   <szobeli_kerdes>Valami7</szobeli_kerdes>
98   <irasbeli_kerdes>Valami4</irasbeli_kerdes>
99 </vizsga>
100
101 <vizsga vizsga ID="3">
102   <I_szam>3800</I_szam>
103   <utca>Fekete</utca>
104   <hazszam>30</hazszam>
105   <gyakorlat>Kinti</gyakorlat>
106   <szobeli_kerdes>Valami3</szobeli_kerdes>
107   <irasbeli_kerdes>Valami8</irasbeli_kerdes>
108 </vizsga>
109
110
111
112
```

```
117 <vizsgalja Vi_Va_Vi="2" Va_Vi_Va="2">
118   <datum>2022.11.16.</datum>
119 </vizsgalja>
120
121 <vizsgalja Vi_Va_Vi="3" Va_Vi_Va="3">
122   <datum>2022.11.23.</datum>
123 </vizsgalja>
124
125
126
127
128 <felugyeli V_F_F="VFF1" V_F_V="VfV1"></felugyeli>
129 <felugyeli V_F_F="VFF2" V_F_V="VfV2"></felugyeli>
130 <felugyeli V_F_F="VFF3" V_F_V="VfV3"></felugyeli>
131
132
133 </vadaszvizsga>
```

1d) Az XML dokumentum alapján XMLSchema készítése (saját típusok, ref, key, keyref, speciális elemek)


```
1. feladat > XMLSchemaDWWUAP.xsd > xs:schema > xs:element > xs:keyref
132 <!-- Idegen kulcsok-->
133
134 <xs:keyref refer="Vadasz_kulcs" name="vadasz_idegen_kulcs">
135 |   <xs:selector xpath="egyesulet" />
136 |   <xs:field xpath="@vadasz-egyesulet" />
137 </xs:keyref>
138
139 <xs:keyref refer="egyesulet_kulcs" name="vizsga_idegen_kulcs">
140 |   <xs:selector xpath="felugyelo" />
141 |   <xs:field xpath="@vadasz-felugyelo" />
142 </xs:keyref>
143
144 <xs:keyref refer="felugyelo_kulcs" name="vizsga2_idegen_kulcs">
145 |   <xs:selector xpath="felugyeli" />
146 |   <xs:field xpath="@felugyelo-vizsga" />
147 </xs:keyref>
148
149 <xs:keyref refer="felugyelo_kulcs" name="vizsga3_idegen_kulcs">
150 |   <xs:selector xpath="vizsga" />
151 |   <xs:field xpath="@Vi_Va_Vi" />
152 </xs:keyref>
153
154 <xs:keyref refer="felugyelo_kulcs" name="vadasz2_idegen_kulcs">
155 |   <xs:selector xpath="vadasz" />
156 |   <xs:field xpath="@Va_Vi_Va" />
157 </xs:keyref>
158
159 <xs:keyref refer="felugyelo_kulcs" name="felugyelo_idegen_kulcs">
160 |   <xs:selector xpath="felugyelo" />
161 |   <xs:field xpath="@V_F_F" />
162 </xs:keyref>
163
164 <xs:keyref refer="vizsga_kulcs" name="vadasz3_idegen_kulcs">
165 |   <xs:selector xpath="vadasz" />
166 |   <xs:field xpath="@V_F_V" />
167 </xs:keyref>
168
169
170 </xs:element>
171
172 </xs:schema>
```

2. feladat

2a) adatolvasás

```

J DOMReadDWWUAP.java X
2. feladat > DomParseDWWUAP > src > hu > domparse > dwwuap > J DOMReadDWWUAP.java > ...
1 package hu.domparse.dwwuap;
2
3 import java.io.File;
4 import java.io.IOException;
5
6 import javax.xml.parsers.DocumentBuilder;
7 import javax.xml.parsers.DocumentBuilderFactory;
8 import javax.xml.parsers.ParserConfigurationException;
9
10 import org.w3c.dom.Document;
11 import org.w3c.dom.Element;
12 import org.w3c.dom.Node;
13 import org.w3c.dom.NodeList;
14 import org.xml.sax.SAXException;
15
16 public class DOMReadDWWUAP {
17
18     Run | Debug
19     public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException {
20         //Forrás file
21         File file = new File(pathname: "XMLDWWUAP.xml");
22
23         DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
24         DocumentBuilder dBuilder = factory.newDocumentBuilder();
25
26         Document doc = dBuilder.parse(file);
27
28         doc.getDocumentElement().normalize();
29         //Gyökérelem
30         System.out.println("Root element: " + doc.getDocumentElement().getNodeName());
31         //Gyerekelemek lementése
32         NodeList nList = (NodeList) doc.getDocumentElement().
33
34         for (int i = 0; i < nList.getLength(); i++) {
35             Node node = nList.item(i);
36
37             //Ha egyesület
38             if (node.getNodeName() == "egyesulet"){
39                 if (!node.getNodeName().equals(anObject: "#text")) {
40                     System.out.println(x: "\n");
41                     System.out.println("Current element: " + node.getNodeName());

```

```

32
33         for (int i = 0; i < nList.getLength(); i++) {
34             Node node = nList.item(i);
35
36
37             //Ha egyesület
38             if (node.getNodeName() == "egyesulet"){
39                 if (!node.getNodeName().equals(anObject: "#text")) {
40                     System.out.println(x: "\n");
41                     System.out.println("Current element: " + node.getNodeName());
42                 }
43                 //Egyesület adatainak kiírása
44                 if (node.getNodeType() == Node.ELEMENT_NODE) {
45                     Element elem = (Element) node;
46
47                     String e_id = elem.getAttribute(name: "E_ID");
48
49                     Node egyesulet_node = elem.getElementsByTagName(name: "telekhely").item(index: 0);
50                     String telekhely_name = egyesulet_node.getTextContent();
51
52                     Node egyesulet_node2 = elem.getElementsByTagName(name: "alapitas_eve").item(index: 0);
53                     String alapitas_eve_name = egyesulet_node2.getTextContent();
54
55                     Node egyesulet_node3 = elem.getElementsByTagName(name: "e_nev").item(index: 0);
56                     String e_nev_name = egyesulet_node3.getTextContent();
57
58                     System.out.printf(format: "Egyesulet id: %s\n", e_id);
59                     System.out.printf(format: "Telekhely neve: %s\n", telekhely_name);
60                     System.out.printf(format: "Alapitas eve: %s\n", alapitas_eve_name);
61                     System.out.printf(format: "Egyesulet neve: %s\n", e_nev_name);
62                 }
63             }
64
65         }
66     }
67 }
68
69 }

```


2b) adatmódosítás

```
J DOMModifyDWWUAP.java X
2. feladat > DomParseDWWUAP > src > hu > domparse > dwwuap > J DOMModifyDWWUAP.java > DOMModifyDWWUAP > main(String[])
1 package hu.domparse.dwwuap;
2
3 import java.io.File;
4 import java.io.IOException;
5
6 import javax.xml.parsers.DocumentBuilder;
7 import javax.xml.parsers.DocumentBuilderFactory;
8 import javax.xml.parsers.ParserConfigurationException;
9 import javax.xml.transform.Transformer;
10 import javax.xml.transform.TransformerFactory;
11 import javax.xml.transform.dom.DOMSource;
12 import javax.xml.transform.stream.StreamResult;
13
14 import org.w3c.dom.Document;
15 import org.w3c.dom.Element;
16 import org.w3c.dom.NamedNodeMap;
17 import org.w3c.dom.Node;
18 import org.w3c.dom.NodeList;
19 import org.xml.sax.SAXException;
20
21 public class DOMModifyDWWUAP {
22
23     Run | Debug
24     public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException {
25         try {
26             //Forrás file
27             File inputFile = new File(pathname: "XML2DWWUAP.xml");
28             DocumentBuilderFactory docFactory = DocumentBuilderFactory.newInstance();
29             DocumentBuilder docBuilder = docFactory.newDocumentBuilder();
30             Document doc = docBuilder.parse(inputFile);
31
32             //Első és harmadik egyesület mentése
33             Node vadasz1 = doc.getElementsByTagName(tagname: "vadasz").item(index: 0);
34             Node vadasz3 = doc.getElementsByTagName(tagname: "vadasz").item(index: 2);
35             //Gyökérelem
36             Node vadaszvizsga = doc.getFirstChild();
37
38             //Harmadik egyesület ID váltás
39             NamedNodeMap attr = vadasz3.getAttributes();
40             Node nodeAttr = attr.getNamedItem(name: "V_ID");
41             nodeAttr.setTextContent(textContent: "20");
```

```

40     Node nodeAttr = attr.getNamedItem(name: "V_ID");
41     nodeAttr.setTextContent(textContent: "20");
42
43     //Második egyesület irányítószámának megváltoztatása
44     NodeList list = vadasz1.getChildNodes();
45     for (int i = 0; i < list.getLength(); i++) {
46         Node node = list.item(i);
47         if (node.getNodeType() == Node.ELEMENT_NODE) {
48             Element elem = (Element) node;
49             if ("telef_szam".equals(elem.getNodeName())) {
50                 elem.setTextContent(textContent: "06206964200");
51             }
52         }
53     }
54
55     //Harmadik egyesületnek az Email megváltoztatása
56     NodeList list1 = vadasz3.getChildNodes();
57     for (int i = 0; i < list1.getLength(); i++) {
58         Node node1 = list1.item(i);
59         if (node1.getNodeType() == Node.ELEMENT_NODE) {
60             Element elem1 = (Element) node1;
61             if ("Email".equals(elem1.getNodeName())) {
62                 elem1.setTextContent(textContent: "valamiModify@Email.com");
63             }
64         }
65     }
66
67     //Egyesületek törlése
68     NodeList childNodes = vadaszvizsga.getChildNodes();
69     for(int i = 0; i < childNodes.getLength(); i++) {
70         Node node = childNodes.item(i);
71
72         if("egyesulet".equals(node.getNodeName()))
73             vadaszvizsga.removeChild(node);
74     }
75
76     //Konzolra kirás
77     TransformerFactory transformerFactory = TransformerFactory.newInstance();
78     Transformer transformer = transformerFactory.newTransformer();
79     DOMSource source = new DOMSource(doc);
80     System.out.println(x: "New File");

```

```

75
76     //Konzolra kirás
77     TransformerFactory transformerFactory = TransformerFactory.newInstance();
78     Transformer transformer = transformerFactory.newTransformer();
79     DOMSource source = new DOMSource(doc);
80     System.out.println(x: "New File");
81     StreamResult consoleResult = new StreamResult(System.out);
82     transformer.transform(source, consoleResult);
83 }catch(Exception e) {
84     e.printStackTrace();
85 }
86
87 }

```

2c) adatlekérdezés

2. feladat > DomParseDWWUAP > src > hu > domparse > dwwuap > J DOMQueryDWWUAP.java > {} hu.domparse.dwwuap

```
1 package hu.domparse.dwwuap;
2
3 import java.io.File;
4 import java.io.IOException;
5
6 import javax.xml.parsers.DocumentBuilder;
7 import javax.xml.parsers.DocumentBuilderFactory;
8 import javax.xml.parsers.ParserConfigurationException;
9
10 import org.w3c.dom.Document;
11 import org.w3c.dom.Element;
12 import org.w3c.dom.Node;
13 import org.w3c.dom.NodeList;
14 import org.xml.sax.SAXException;
15
16 public class DOMQueryDWWUAP {
17
18     Run | Debug
19     public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException {
20         //Forrás file
21         File file = new File(pathname: "XMLDWWUAP.xml");
22
23         DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
24         DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
25
26         Document doc = dBuilder.parse(file);
27         doc.getDocumentElement().normalize();
28         //Gyökér elem
29         System.out.print(s: "Root element: ");
30         System.out.println(doc.getDocumentElement().getNodeName());
31         //Vadaszvizsga betöltése
32         NodeList nList = doc.getElementsByTagName(tagname: "vadaszvizsga");
33
34         System.out.println(x: "-----");
35
36         //Végigfut a vadasznak a gyerek elemein kihagyva a kort
37         for (int i = 0; i < nList.getLength(); i++) {
38             Node node = nList.item(i);
39             System.out.println("\nCurrent Element : "+node.getNodeName());
40             if (node.getNodeType() == Node.ELEMENT_NODE) {
41                 Element elem = (Element) node;
42                 System.out.println("ID: "+elem.getAttribute(name: "V ID"));
```

```

24
25 Document doc = dBuilder.parse(file);
26 doc.getDocumentElement().normalize();
27 //Gyökér elem
28 System.out.print(s: "Root element: ");
29 System.out.println(doc.getDocumentElement().getNodeName());
30 //Vadaszvizsga betöltése
31 NodeList nList = doc.getElementsByTagName(tagname: "vadaszvizsga");
32
33 System.out.println(x: "-----");
34
35 //Végigfut a vadasznak a gyerek elemein kihagyva a kort
36 for (int i = 0; i < nList.getLength(); i++) {
37     Node node = nList.item(i);
38     System.out.println("\nCurrent Element : "+node.getNodeName());
39     if (node.getNodeType() == Node.ELEMENT_NODE) {
40         Element elem = (Element) node;
41         System.out.println("ID:"+elem.getAttribute(name: "V_ID"));
42         NodeList nList2 = elem.getChildNodes();
43         for (int j = 0; j < nList2.getLength(); j++) {
44             Node node2 = nList2.item(j);
45             if (node2.getNodeType() == Node.ELEMENT_NODE) {
46                 Element elem2 = (Element) node2;
47                 if(!node2.getNodeName().equals(anObject: "kor")) {
48                     System.out.println(node2.getNodeName()+" : "+node2.getTextContent());
49                 }
50                 NodeList nList3 = elem2.getChildNodes();
51                 for (int k = 0; k < nList3.getLength(); k++) {
52                     Node node3 = nList3.item(k);
53                     if(node3.getNodeType()==Node.ELEMENT_NODE) {
54                         System.out.println("vadaszvizsga : "+node3.getNodeName()+" : "+node3.getTextContent());
55                     }
56                 }
57             }
58         }
59     }
60 }
61 }
62 }

```