

# **Jegyzőkönyv**

Adatkezelés XML-ben

Féléves feladat

Készítette: **Csurgai Ádám János**

Neptunkód: **VU2RVC**

# 1. Feladat

## A feladat leírása

A feladatban egy Szóbeli érettségi adatbázisának ER-modelljét készítem el, majd átalakítom relációs adatmodellre.

A **Szóbeli érettségi** egyedhez tartoznak az *ID*, az *Időpont*, a *Helyszín*, és a *Szint* tulajdonság. A *Szint* azt mondja meg, hogy emelt, vagy középszintű az adott **Szóbeli érettségi**. A *Tantárgy* többértékű tulajdonság, mivel az érettségien akár több tárgyból is lehet szóbelizni. A *Kód* (Egyedi, számjegyekből álló azonosító) kulcsként funkcionál, mert ez nyilvánvalóan beazonosítja a **Szóbeli érettségit**.

A **Tanár** egyed *ID*, *Tanított tárgy*, valamint egy összetett *Név* tulajdonságból áll, amelyhez a *Vezetéknév* és az *Utónév* tag tartozik. Az *Azonosító kód* (Egyedi, számjegyekből álló azonosító) a kulcs, amely beazonosít.

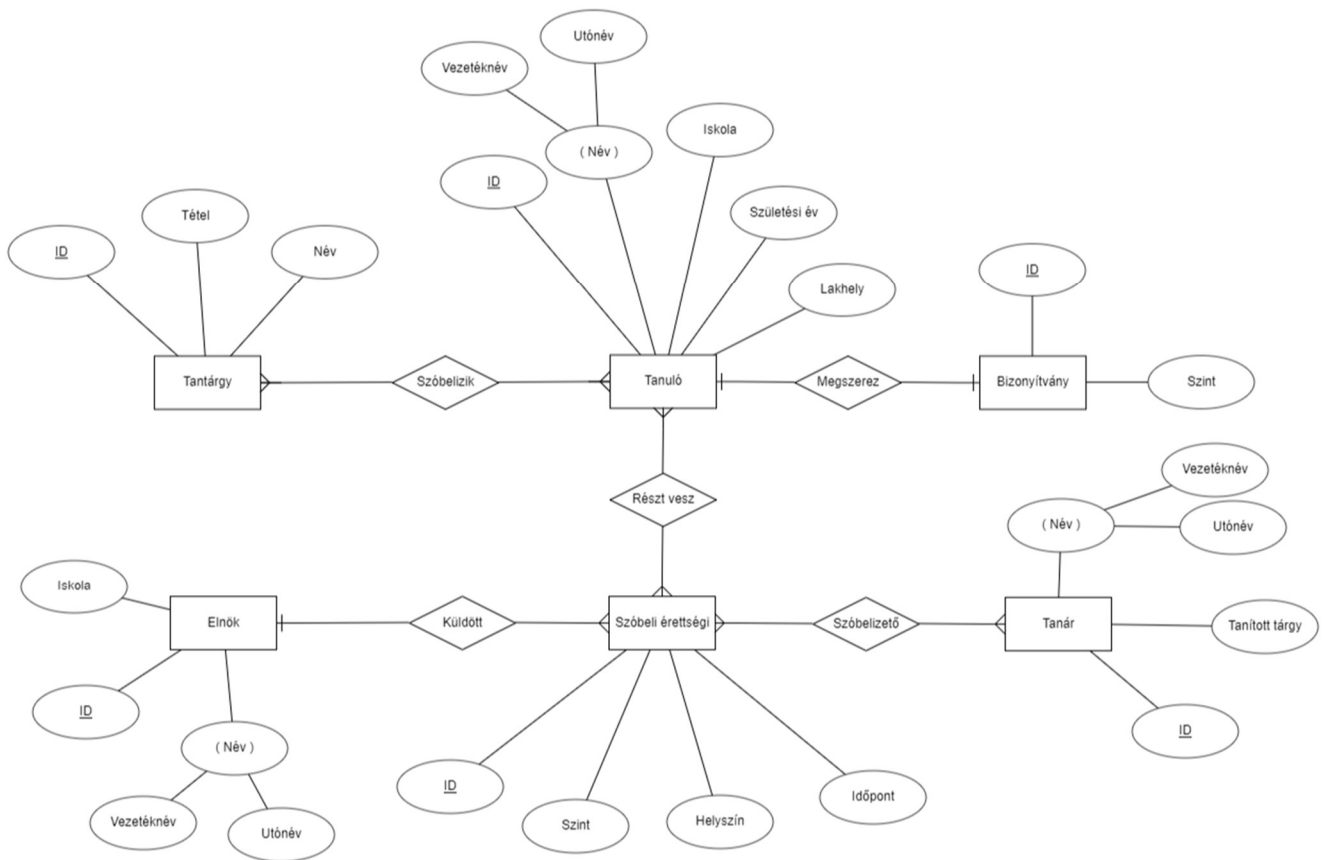
Az **Elnök** egyed *ID kódja* (Egyedi, számjegyekből álló azonosító) a kulcs. Az *Iskola* azt mutatja, hogy honnan jött az **Elnök**. Továbbá van egy *Név* összetett tulajdonság, mely *Vezetéknév* és *Utónév* tagokból áll.

A **Tanuló** egyed *ID-ja* (Egyedi, számjegyekből álló azonosító) a kulcs, mely beazonosítja az adott **Tanulót**. A *Név* attribútum egy összetett tulajdonság, melynek tagjai a *Vezetéknév* és az *Utónév*. Az *Iskola* tulajdonság azt mondja meg, hogy melyik iskolában tanul a diák. További tulajdonság: *Lakhely*.

A **Bizonyítvány** egyednek az *ID* attribútuma a kulcs, amely beazonosítja az egyedet. A **Bizonyítványhoz** továbbá tartozik egy *Szint* tulajdonság.

A **Tantárgy** egyed tulajdonságai az *ID* (amely a tárgy kulcsa), a *Név* és a *Tétel*.

## a) Az adatbázis ER modellje



## b) Az ER modell konvertálása XDM modellre

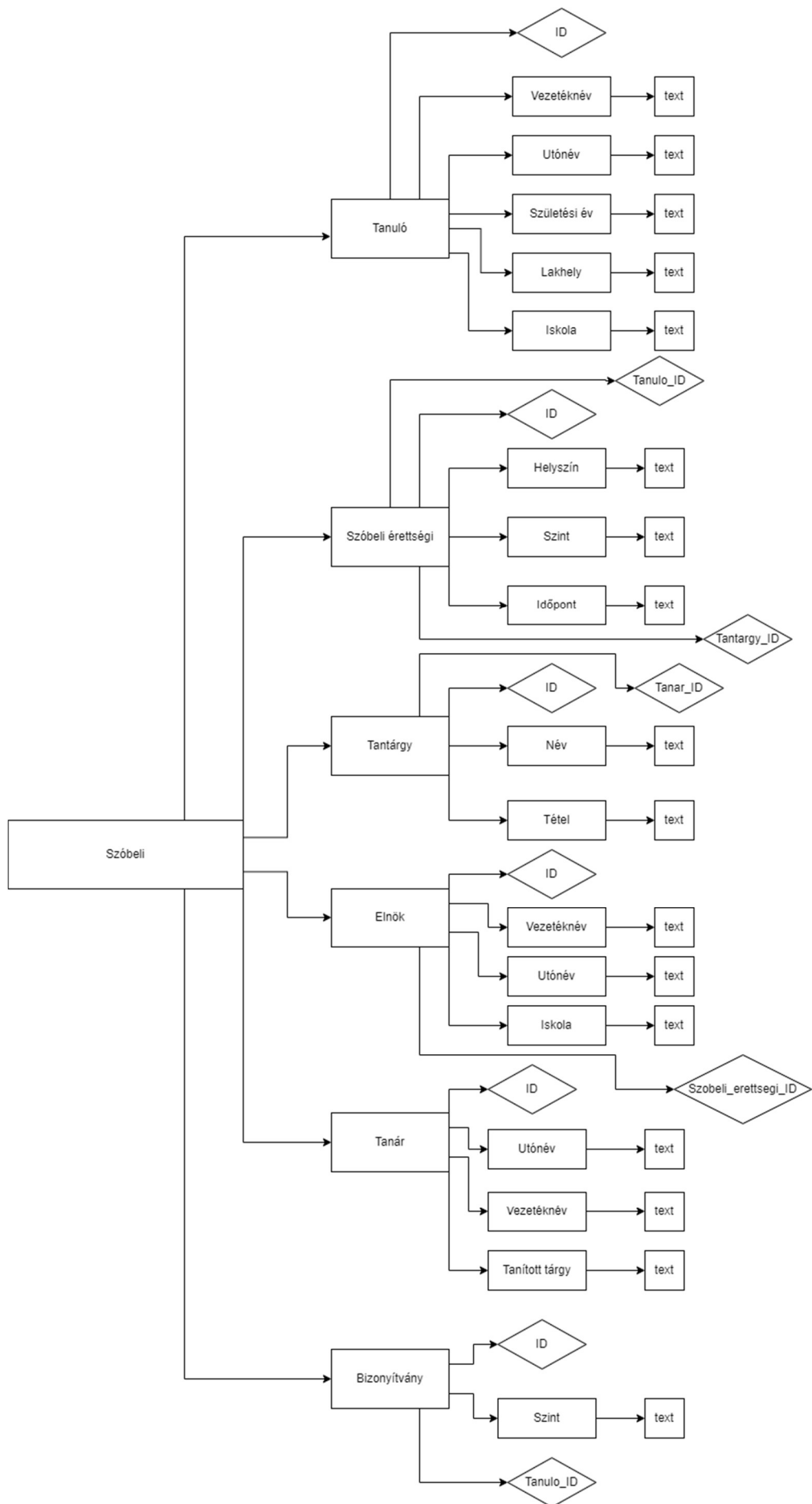
A konvertálás következtében több idegen kulcs is létrejön:

A **Szóbeli érettségi** egyedhez egy *tanulo\_ID* és egy *Tantagy\_ID* idegen kulcs jön létre, mely a **Tanuló**ra és a **Tantárgy**ra mutat.

A **tantárgy**nál egy *tanar\_ID* idegen kulcs jön létre, amely a **Tanár**ra mutat.

Az **Elnök**nél egy *szobeli\_erettsegi\_ID* idegen kulcs jön létre, amely a **Szóbeli érettségire** mutat.

A bizonyítványnál egy *tanulo\_ID* jön létre, amely a **Tanuló**ra mutat.



## c) Az XML dokumentum

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<szobeli xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="XSDVU2RVC.xsd">
  <tanulo id="01">
    <vezeteknev>Hurka</vezeteknev>
    <utonev>Gyurka</utonev>
    <szuletési_ev>1999</szuletési_ev>
    <lakhely>Miskolc</lakhely>
    <iskola>Bláthy</iskola>
  </tanulo>
  <tanulo id="02">
    <vezeteknev>Bármi</vezeteknev>
    <utonev>Áron</utonev>
    <szuletési_ev>1995</szuletési_ev>
    <lakhely>Miskolc</lakhely>
    <iskola>Diósgyőri</iskola>
  </tanulo>
  <tanulo id="03">
    <vezeteknev>Ebéd</vezeteknev>
    <utonev>Elek</utonev>
    <szuletési_ev>2001</szuletési_ev>
    <lakhely>Miskolc</lakhely>
    <iskola>Avasi</iskola>
  </tanulo>
  <szobeli_erettsegi id="01" tantargy_ID="01" tanulo_ID="01">
    <helyszin>Bláthy</helyszin>
    <szint>közép</szint>
    <idopont>2016-06-15</idopont>
  </szobeli_erettsegi>
  <szobeli_erettsegi id="02" tantargy_ID="02" tanulo_ID="02">
    <helyszin>Fényi</helyszin>
    <szint>emelt</szint>
    <idopont>2018-06-19</idopont>
  </szobeli_erettsegi>
  <szobeli_erettsegi id="03" tantargy_ID="03" tanulo_ID="03">
    <helyszin>Földes</helyszin>
    <szint>közép</szint>
    <idopont>2012-07-01</idopont>
  </szobeli_erettsegi>
  <tantargy id="01" tanar_ID="01">
    <nev>angol</nev>
    <tetel>12</tetel>
  </tantargy>
  <tantargy id="02" tanar_ID="03">
    <nev>Történelem</nev>
    <tetel>6</tetel>
  </tantargy>
  <tantargy id="03" tanar_ID="02">
    <nev>Informatika</nev>
    <tetel>8</tetel>
  </tantargy>
</szobeli>
```

```
</tantargy>
<elnok id="01" szobeli_ID="01">
  <vezeteknev>Dil</vezeteknev>
  <utonev>Emma</utonev>
  <iskola>Zrínyi</iskola>
</elnok>
<elnok id="02" szobeli_ID="02">
  <vezeteknev>Para</vezeteknev>
  <utonev>Zita</utonev>
  <iskola>Diósgyőri</iskola>
</elnok>
<elnok id="03" szobeli_ID="03">
  <vezeteknev>Riz</vezeteknev>
  <utonev>Ottó</utonev>
  <iskola>Avasi</iskola>
</elnok>
<tanar id="01">
  <vezeteknev>Kala</vezeteknev>
  <utonev>Pál</utonev>
  <tanitott_targy>angol</tanitott_targy>
</tanar>
<tanar id="02">
  <vezeteknev>Major</vezeteknev>
  <utonev>Anna</utonev>
  <tanitott_targy>Informatika</tanitott_targy>
</tanar>
<tanar id="03">
  <vezeteknev>Fejet</vezeteknev>
  <utonev>Lenke</utonev>
  <tanitott_targy>Történelem</tanitott_targy>
</tanar>
<bizonyitvany id="01" tanulo_ID="01">
  <szint>közép</szint>
</bizonyitvany>
<bizonyitvany id="02" tanulo_ID="02">
  <szint>emelt</szint>
</bizonyitvany>
<bizonyitvany id="03" tanulo_ID="03">
  <szint>közép</szint>
</bizonyitvany>
</szobeli>
```

## d) Az XSD modell

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

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="szobeli">
    <xs:complexType>
      <xs:sequence>
        <xs:choice maxOccurs="unbounded">
          <xs:element name="tanulo" type="tanulo_type"/>
          <xs:element name="szobeli_erettsegi" type="szobeli_erettsegi_type"/>
          <xs:element name="tantargy" type="tantargy_type"/>
          <xs:element name="elnok" type="elnok_type"/>
          <xs:element name="tanar" type="tanar_type"/>
          <xs:element name="bizonyitvany" type="bizonyitvany_type"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="tanulo_type">
    <xs:sequence>
      <xs:element name="vezeteknev" type="xs:string"/>
      <xs:element name="utonev" type="xs:string"/>
      <xs:element name="szuletesi_ev" type="xs:integer"/>
      <xs:element name="lakhely" type="xs:string"/>
      <xs:element name="iskola" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:unsignedByte"/>
  </xs:complexType>
  <xs:complexType name="szobeli_erettsegi_type">
    <xs:sequence>
      <xs:element name="helyszin" type="xs:string"/>
      <xs:element name="szint" type="xs:string"/>
      <xs:element name="idopont" type="xs:date"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:unsignedByte"/>
    <xs:attribute name="tanulo_ID" type="xs:unsignedByte"/>
    <xs:attribute name="tantargy_ID" type="xs:unsignedByte"/>
  </xs:complexType>
  <xs:complexType name="tantargy_type">
    <xs:sequence>
      <xs:element name="nev" type="xs:string"/>
      <xs:element name="tetel" type="xs:integer"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:unsignedByte"/>
    <xs:attribute name="tanar_ID" type="xs:unsignedByte"/>
  </xs:complexType>
</xs:schema>
```

```
</xs:complexType>
<xs:complexType name="elnok_type">
  <xs:sequence>
    <xs:element name="vezeteknev" type="xs:string"/>
    <xs:element name="utonev" type="xs:string"/>
    <xs:element name="iskola" type="xs:string"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:unsignedByte"/>
  <xs:attribute name="szobeli_ID" type="xs:unsignedByte"/>
</xs:complexType>
<xs:complexType name="tanar_type">
  <xs:sequence>
    <xs:element name="vezeteknev" type="xs:string"/>
    <xs:element name="utonev" type="xs:string"/>
    <xs:element name="tanitott_targy" type="xs:string"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:unsignedByte"/>
</xs:complexType>
<xs:complexType name="bizonyitvany_type">
  <xs:sequence>
    <xs:element name="szint" type="xs:string"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:unsignedByte"/>
  <xs:attribute name="tanulo_ID" type="xs:unsignedByte"/>
</xs:complexType>
</xs:schema>
```



## 2.

### a) DOMRead

```
1 package hu.domparsing.VU2RVC;
2
3 import java.io.File;
4
5 import javax.xml.parsers.DocumentBuilderFactory;
6 import javax.xml.parsers.DocumentBuilder;
7
8 import org.w3c.dom.Document;
9 import org.w3c.dom.NodeList;
10 import org.w3c.dom.Node;
11 import org.w3c.dom.Element;
12
13 public class DomReadVU2RVC {
14
15     public static void main(String[] args){
16
17         try {
18             File inputFile = new File("G:\\Ádám\\Egyetem\\XML\\Beadando\\beadando\\XMLVU2RVC.xml");
19
20             //document Builder
21             DocumentBuilderFactory dbFactory
22                 = DocumentBuilderFactory.newInstance();
23             DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
24             Document doc = dBuilder.parse(inputFile);
25             doc.getDocumentElement().normalize();
26             System.out.println("Root element : "
27                 + doc.getDocumentElement().getNodeName());
28
29             //tanuló
30             NodeList nList = doc.getElementsByTagName("tanulo");
31             System.out.println("-----");
32
33             //List every data from the tanulo element
34             for (int temp = 0; temp < nList.getLength(); temp++) {
35                 Node nNode = nList.item(temp);
36                 System.out.println("\nCurrent Element : "
37                     + nNode.getNodeName());
38                 if (nNode.getNodeType() == Node.ELEMENT_NODE) {
39                     Element eElement = (Element) nNode;
40                     System.out.println("ID: "
41                         + eElement.getAttribute("id"));
42                     System.out.println("Vezetéknév : "
43                         + eElement
44                             .getElementsByTagName("vezeteknev")
45                             .item(0)
46                             .getTextContent());
47                     System.out.println("Utónév : "
48                         + eElement
49                             .getElementsByTagName("utonev")
50                             .item(0)
51                             .getTextContent());
```

```

52     System.out.println("Születési év: "
53     + eElement
54         .getElementsByTagName("szuletesi_ev")
55         .item(0)
56         .getTextContent());
57     System.out.println("Lakhely: "
58     + eElement
59         .getElementsByTagName("lakhely")
60         .item(0)
61         .getTextContent());
62     System.out.println("Iskola: "
63     + eElement
64         .getElementsByTagName("iskola")
65         .item(0)
66         .getTextContent());
67 }
68 }
69 //szóbeli érettségi
70 nList = doc.getElementsByTagName("szobeli_erettsegi");
71 System.out.println("-----");
72 //List every data from the szobeli erettsegi element
73 for (int temp = 0; temp < nList.getLength(); temp++) {
74     Node nNode = nList.item(temp);
75     System.out.println("\nCurrent Element : "
76     + nNode.getNodeName());
77     if (nNode.getNodeType() == Node.ELEMENT_NODE) {
78         Element eElement = (Element) nNode;
79         System.out.println("ID: "
80         + eElement.getAttribute("id"));
81         System.out.println("Helyszín: "
82         + eElement
83             .getElementsByTagName("helyszin")
84             .item(0)
85             .getTextContent());
86         System.out.println("Szint: "
87         + eElement
88             .getElementsByTagName("szint")
89             .item(0)
90             .getTextContent());
91         System.out.println("Időpont: "
92         + eElement
93             .getElementsByTagName("idopont")
94             .item(0)
95             .getTextContent());
96     }
97 }
98 //Tantárgy
99 nList = doc.getElementsByTagName("tantargy");
100 System.out.println("-----");
101 //List every data from the tantargy element
102 for (int temp = 0; temp < nList.getLength(); temp++) {
103     Node nNode = nList.item(temp);

```

```

104         System.out.println("\nCurrent Element : "
105             + nNode.getNodeName());
106         if (nNode.getNodeType() == Node.ELEMENT_NODE) {
107             Element eElement = (Element) nNode;
108             System.out.println("ID: "
109                 + eElement.getAttribute("id"));
110             System.out.println("Név: "
111                 + eElement
112                     .getElementsByTagName("nev")
113                     .item(0)
114                     .getTextContent());
115             System.out.println("Tétel: "
116                 + eElement
117                     .getElementsByTagName("tetel")
118                     .item(0)
119                     .getTextContent());
120         }
121     }
122     //Elnök
123     nList = doc.getElementsByTagName("elnok");
124     System.out.println("-----");
125     //List every data from the elnok element
126     for (int temp = 0; temp < nList.getLength(); temp++) {
127         Node nNode = nList.item(temp);
128         System.out.println("\nCurrent Element : "
129             + nNode.getNodeName());
130         if (nNode.getNodeType() == Node.ELEMENT_NODE) {
131             Element eElement = (Element) nNode;
132             System.out.println("ID: "
133                 + eElement.getAttribute("id"));
134             System.out.println("Vezetéknév: "
135                 + eElement
136                     .getElementsByTagName("vezeteknev")
137                     .item(0)
138                     .getTextContent());
139             System.out.println("Utónév: "
140                 + eElement
141                     .getElementsByTagName("utonev")
142                     .item(0)
143                     .getTextContent());
144             System.out.println("Iskola: "
145                 + eElement
146                     .getElementsByTagName("iskola")
147                     .item(0)
148                     .getTextContent());
149         }
150     }
151 }
152 //Tanár
153 nList = doc.getElementsByTagName("tanar");
154 System.out.println("-----");

```

```

155 //List every data from the tanac element
156 for (int temp = 0; temp < nList.getLength(); temp++) {
157     Node nNode = nList.item(temp);
158     System.out.println("\nCurrent Element : "
159         + nNode.getNodeName());
160     if (nNode.getNodeType() == Node.ELEMENT_NODE) {
161         Element eElement = (Element) nNode;
162         System.out.println("ID: "
163             + eElement.getAttribute("id"));
164         System.out.println("Vezetéknév: "
165             + eElement
166                 .getElementsByTagName("vezeteknev")
167                 .item(0)
168                 .getTextContent());
169         System.out.println("Utónév: "
170             + eElement
171                 .getElementsByTagName("utonev")
172                 .item(0)
173                 .getTextContent());
174         System.out.println("Tanított tárgy: "
175             + eElement
176                 .getElementsByTagName("tanitott_targy")
177                 .item(0)
178                 .getTextContent());
179     }
180 }
181
182 //Bizonyítvány
183 nlist = doc.getElementsByTagName("bizonyitvany");
184 System.out.println("-----");
185 //List every data from the bizonyitvany element
186 for (int temp = 0; temp < nList.getLength(); temp++) {
187     Node nNode = nList.item(temp);
188     System.out.println("\nCurrent Element : "
189         + nNode.getNodeName());
190     if (nNode.getNodeType() == Node.ELEMENT_NODE) {
191         Element eElement = (Element) nNode;
192         System.out.println("ID: "
193             + eElement.getAttribute("id"));
194         System.out.println("Szint: "
195             + eElement
196                 .getElementsByTagName("szint")
197                 .item(0)
198                 .getTextContent());
199     }
200 }
201
202 } catch (Exception e) {
203     e.printStackTrace();
204 }
205 }
206
207 }

```

## b) DOMQuery

```
1 package hu.domparse.VU2RVC;
2
3 import javax.xml.parsers.DocumentBuilderFactory;
4 import javax.xml.parsers.DocumentBuilder;
5 import org.w3c.dom.Document;
6 import org.w3c.dom.NodeList;
7 import org.w3c.dom.Node;
8 import org.w3c.dom.Element;
9 import java.io.File;
10
11 //this helped me
12 //https://www.tutorialspoint.com/java_xml/java_dom_query_document.htm
13
14
15 public class DOMQueryVU2RVC {
16     public static final String xmlFilePath = "G:\\XML\\Beadando\\beadando\\XMLVU2RVC.xml";
17
18     public static void main(String argv[]) {
19
20         try {
21             File inputFile = new File(xmlFilePath);
22             //document builder
23             DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
24             DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
25             Document doc = dBuilder.parse(inputFile);
26             doc.getDocumentElement().normalize();
27             System.out.print("Root element: ");
28             System.out.println(doc.getDocumentElement().getNodeName());
29             //selecting the elements
30             NodeList nList = doc.getElementsByTagName("szobeli_erejtsegi");
31             NodeList mList = doc.getElementsByTagName("tanulo");
32             NodeList kList = doc.getElementsByTagName("tantargy");
33             NodeList lList = doc.getElementsByTagName("tanar");
34
35             System.out.println("-----");
36
37             for (int temp = 0; temp < nList.getLength(); temp++) {
38                 Node nNode = nList.item(temp);
39                 System.out.println("\nCurrent Element :");
40                 System.out.print(nNode.getNodeName());
41
42                 if (nNode.getNodeType() == Node.ELEMENT_NODE) {
43                     Element eElement = (Element) nNode;
44                     System.out.print("\nID : ");
45                     System.out.println(eElement.getAttribute("id"));
46                     //selecting the idopont element from the given (szobeli_erejtsegi) element
47                     NodeList idopontList = eElement.getElementsByTagName("idopont");
48
49                     //List all idopont element, and ID-s
50                     for (int count = 0; count < idopontList.getLength(); count++) {
51                         Node node1 = idopontList.item(count);
52
53                     }
```



```

54
55         if (node1.getNodeType() == Node.ELEMENT_NODE) {
56             Element idoPont = (Element) node1;
57             System.out.println("Időpont : ");
58             System.out.println(idoPont.getTextContent());
59             System.out.println("Tanuló ID : ");
60             System.out.println(eElement.getAttribute("tanulo_ID"));
61             System.out.println("Tantárgy ID : ");
62             System.out.println(eElement.getAttribute("tantargy_ID"));
63         }
64     }
65 }
66 }
67 }
68 for (int temp = 0; temp < mList.getLength(); temp++) {
69     Node nNode = mList.item(temp);
70     System.out.println("\nCurrent Element :");
71     System.out.print(nNode.getNodeName());
72
73     if (nNode.getNodeType() == Node.ELEMENT_NODE) {
74         Element eElement = (Element) nNode;
75         System.out.print("\nID : ");
76         System.out.println(eElement.getAttribute("id"));
77         NodeList vezetékList = eElement.getElementsByTagName("vezeteknev");
78         NodeList keresztnévList = eElement.getElementsByTagName("utonev");
79         NodeList születettList = eElement.getElementsByTagName("szuletesi_ev");
80
81         for (int count = 0; count < vezetékList.getLength(); count++) {
82             Node node1 = vezetékList.item(count);
83             Node node2 = keresztnévList.item(count);
84
85             if (node1.getNodeType() == Node.ELEMENT_NODE && node2.getNodeType() == Node.ELEMENT_NODE) {
86                 Element vezeték = (Element) node1;
87                 Element keresztnév = (Element) node2;
88                 System.out.print("Név : ");
89                 System.out.println(vezeték.getTextContent() + " " + keresztnév.getTextContent());
90             }
91         }
92         for (int count = 0; count < születettList.getLength(); count++) {
93             Node node1 = születettList.item(count);
94
95             if (node1.getNodeType() == Node.ELEMENT_NODE) {
96                 Element született = (Element) node1;
97                 System.out.print("Születési év : ");
98                 System.out.println(született.getTextContent());
99             }
100         }
101     }
102 }

```

```

103
104 for (int temp = 0; temp < kList.getLength(); temp++) {
105     Node nNode = kList.item(temp);
106     System.out.println("\nCurrent Element :");
107     System.out.print(nNode.getNodeName());
108
109     if (nNode.getNodeType() == Node.ELEMENT_NODE) {
110         Element eElement = (Element) nNode;
111         System.out.print("\nID : ");
112         System.out.println(eElement.getAttribute("id"));
113         NodeList névList = eElement.getElementsByTagName("nev");
114
115         for (int count = 0; count < névList.getLength(); count++) {
116             Node node1 = névList.item(count);
117
118             if (node1.getNodeType() == Node.ELEMENT_NODE) {
119                 Element név = (Element) node1;
120                 System.out.print("Tantárgy : ");
121                 System.out.println(név.getTextContent());
122                 System.out.print("Tanár ID : ");
123                 System.out.println(eElement.getAttribute("tanar_ID"));
124             }
125         }
126     }
127 }
128 for (int temp = 0; temp < lList.getLength(); temp++) {
129     Node nNode = lList.item(temp);
130     System.out.println("\nCurrent Element :");
131     System.out.print(nNode.getNodeName());
132
133     if (nNode.getNodeType() == Node.ELEMENT_NODE) {
134         Element eElement = (Element) nNode;
135         System.out.print("\nID : ");
136         System.out.println(eElement.getAttribute("id"));
137         NodeList vezetékList = eElement.getElementsByTagName("vezeteknev");
138         NodeList keresztnévList = eElement.getElementsByTagName("utonev");

```

```

138
139
140         for (int count = 0; count < vezetekList.getLength(); count++) {
141             Node node1 = vezetekList.item(count);
142             Node node2 = keresztList.item(count);
143
144             if (node1.getNodeType() == Node.ELEMENT_NODE && node2.getNodeType() == Node.ELEMENT_NODE) {
145                 Element vezetek = (Element) node1;
146                 Element kereszt = (Element) node2;
147                 System.out.print("Név : ");
148                 System.out.println(vezetek.getTextContent()+" "+ kereszt.getTextContent());
149             }
150         }
151     }
152
153
154     } catch (Exception e) {
155         e.printStackTrace();
156     }
157 }
158
159 }
160

```

## c) DOMModify

```
1 package hu.dompase.VU2RVC;
2
3 import java.io.File;
4
19
20
21 public class DOMModifyVU2RVC {
22
23     public static final String xmlFilePath = "G:\\XML\\Beadando\\beadando\\XMLVU2RVC.xml";
24
25     public static void main(String argv[]) {
26
27         try {
28
29             //document builder
30             DocumentBuilderFactory documentBuilderFactory = DocumentBuilderFactory.newInstance();
31
32             DocumentBuilder documentBuilder = documentBuilderFactory.newDocumentBuilder();
33
34             Document document = documentBuilder.parse(xmlFilePath);
35
36             // Get bizonvitany by tag name
37             //getting the third bizonvitany
38             Node bizonvitany = document.getElementsByTagName("bizonvitany").item(2);
39
40             // update bizonvitany elements id
41             NamedNodeMap attribute = bizonvitany.getAttributes();
42             Node nodeAttr = attribute.getNamedItem("id");
43             nodeAttr.setTextContent("03");
44
45             // append a new node to the first bizonvitany
46             Element succes = document.createElement("atment");
47
48             succes.appendChild(document.createTextNode("igen"));
49
50             bizonvitany.appendChild(succes);
51
52             // loop the bizonvitany node and update salary value, and delete a node
53             NodeList nodes = bizonvitany.getChildNodes();
54
55             for (int i = 0; i < nodes.getLength(); i++) {
56
57                 Node element = nodes.item(i);
58
59                 if ("szint".equals(element.getNodeName())) {
60                     element.setTextContent("közép");
61                 }
62
63                 // remove the succes element
64                 if ("atment".equals(element.getNodeName())) {
65                     bizonvitany.removeChild(element);
66                 }
67
68             }
69
70             // write the DOM object to the file
71             TransformerFactory transformerFactory = TransformerFactory.newInstance();
72
73             Transformer transformer = transformerFactory.newTransformer();
74             DOMSource domSource = new DOMSource(document);
75
76             StreamResult streamResult = new StreamResult(new File(xmlFilePath));
77             transformer.transform(domSource, streamResult);
78
79             System.out.println("The XML File was modified");
80
81             } catch (ParserConfigurationException pce) {
82                 pce.printStackTrace();
83             } catch (TransformerException tfe) {
84                 tfe.printStackTrace();
85             } catch (IOException ioe) {
86                 ioe.printStackTrace();
87             } catch (SAXException sae) {
88                 sae.printStackTrace();
89             }
90         }
91     }
92 }
```