

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Készítette: **Virág Szabolcs**

Neptunkód: **DO425G**

A feladat leírása:

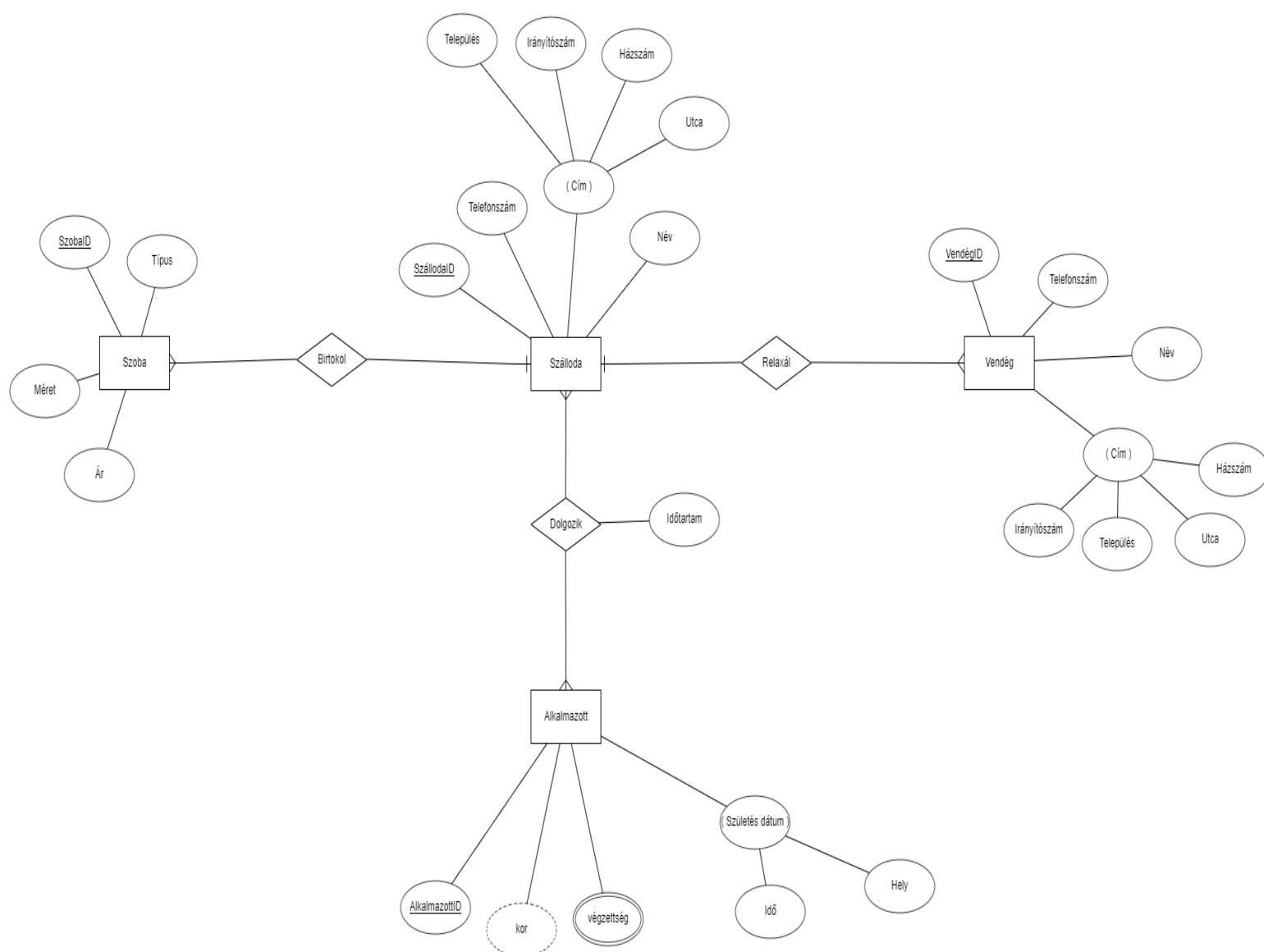
A feladatban egy szálloda adatbázisát készítettem el, amelyben nyilvántartjuk a szálloda vendégeinek, alkalmazottainak, és szobáinak tulajdonságait, adatait.

Az adatbázis az alábbiakat tartalmazza:

A szálloda egyed a szálloda tulajdonságait tartalmazza, köztük a SzállodaID, telefonszám, név, és a cím. A cím összetett tulajdonság, település, irányítószám, házszám, és utca tulajdonságokat foglalja magába. A szoba egyed a szálloda szobáit mutatja be, hogy milyen különböző típusokhoz milyen szoba méret társul és mennyibe kerülnek. Az alkalmazott egyed a szállodában dolgozók adatait tárolják, ahol a kor egy származtatott tulajdonság amely a születési dátumból kiszámítható és a végzettség többértékű tulajdonság mert 1 alkalmazott rendelkezhet több végzettséggel is. A vendég egyed a szállodába érkező vendégek adatait tárolja. A vendég és szálloda egyed között 1:N kapcsolat van mert egy szállodának lehet több vendége viszont egy vendég csak egy szállodában tölti az idejét. Az alkalmazott és a szálloda egyed között N:M kapcsolat van mert egy szállodába több alkalmazott is dolgozik, és egy alkalmazott több szállodába is dolgozhat. A szoba és szálloda egyed között 1:N kapcsolat van mert egy szállodához több szoba tartozik, de egy szoba csak 1 szállodához kapcsolódhat. Az alkalmazottak munkaidejét a dolgozik kapcsolat, időtartam tulajdonsága tárolja.

1.feladat

1a) Az adatbázis ER modell:

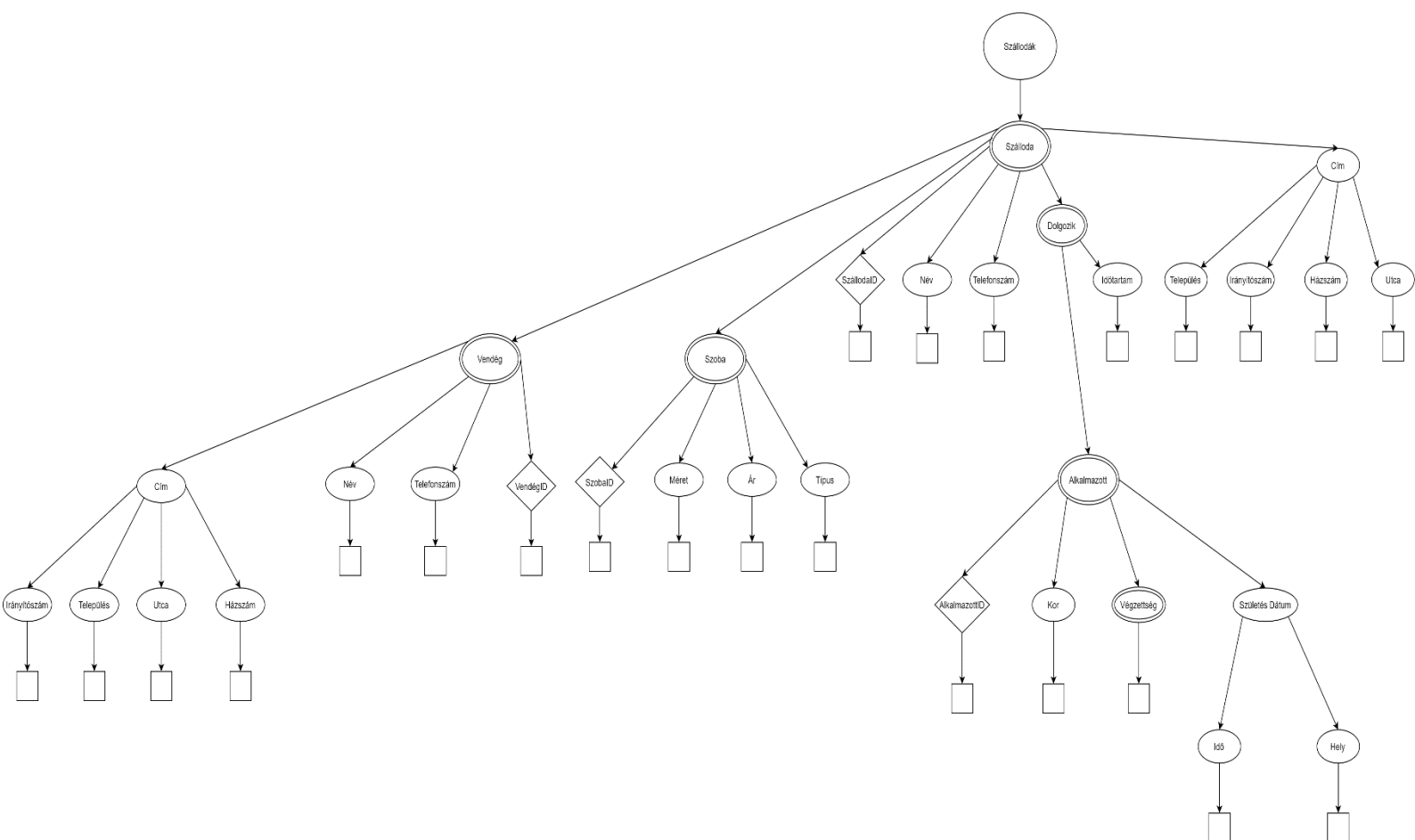


1b) Az adatbázis konvertálása XDM modellre: Az XDM modell a hierarchia leírására szolgál, az XML dokumentumot fa alakban írja le. Alapvetően előfordulást ír le, nem sémát. Építő kövei:

- Elem (ellipszis),
- Szöveg (téglalap),
- Attribútum (rombusz).

Az XDM kiegészítése séma elemekkel:

- Többszörös előfordulás (dupla ellipszis),
- Opcionális elem (ellipszis szaggatott vonallal),
- Kötelező elem (ellipszis vastag vonallal),
- Elsődleges kulcs (attribútum neve aláhúzva),
- Idegen kulcs (attribútum neve aláhúzva szaggatott vonallal).



1c) Az XDM modell alapján XML dokumentum készítése:

```
1. <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2. <Szallodak xmlns:xsi="http://www.w3.org/2001/XMLSchema-
   instance" xsi:noNamespaceSchemaLocation="XMLSchemaD0425G.xsd">
3.   <szalloda ref_alkalmazott="135792CA" ref_szoba="100" ref_vendeg="123456AB" szall
   oda_adoszam="12345678-2-25">
4.     <nev>Teszt 1</nev>
5.     <cim>
6.       <irsz>3700</irsz>
7.       <telepules>Kazincbarcika</telepules>
8.       <utca>Építők utca</utca>
9.       <hsz>44</hsz>
10.    </cim>
11.    <telefonszam>302345678</telefonszam>
12.  </szalloda>
13.
14.  <szalloda ref_alkalmazott="246819MA" ref_szoba="101" ref_vendeg="123456AB" szall
   oda_adoszam="87654321-2-25">
15.    <nev>Teszt 2</nev>
16.    <cim>
17.      <irsz>3519</irsz>
18.      <telepules>Miskolc</telepules>
19.      <utca>Teszt utca</utca>
20.      <hsz>69</hsz>
21.    </cim>
22.    <telefonszam>202345678</telefonszam>
23.  </szalloda>
24.
25.  <szalloda ref_alkalmazott="147259GB" ref_szoba="102" ref_vendeg="245246AB" szall
   oda_adoszam="54654754-2-25">
26.    <nev>Teszt 3</nev>
27.    <cim>
28.      <irsz>4400</irsz>
29.      <telepules>Nyíregyháza</telepules>
30.      <utca>Dísz tér</utca>
31.      <hsz>48</hsz>
32.    </cim>
33.    <telefonszam>702345678</telefonszam>
34.  </szalloda>
35.
36.  <szoba ref_szalloda="12345678-2-25" szoba_ajtoszam="101">
37.    <meret>42</meret>
38.    <ar>31500</ar>
39.    <tipus>superior</tipus>
40.  </szoba>
41.
42.  <szoba ref_szalloda="12345678-2-25" szoba_ajtoszam="100">
43.    <meret>18</meret>
44.    <ar>15000</ar>
45.    <tipus>stantard</tipus>
46.  </szoba>
47.
48.  <szoba ref_szalloda="12345678-2-25" szoba_ajtoszam="102">
49.    <meret>18</meret>
50.    <ar>15000</ar>
51.    <tipus>duplex</tipus>
52.  </szoba>
53.
54.
55.  <vendeg ref_szalloda="54654754-2-25" vendeg_szigsz="245246AB">
56.    <nev>
57.      <vendegnev>Nagy Tibor</vendegnev>
58.    </nev>
59.  </cim>
```

```

60.         <irsz>3519</irsz>
61.         <telepules>Miskolc</telepules>
62.         <utca>Vadgalamb utca</utca>
63.         <hsz>21</hsz>
64.     </cim>
65.     <telefonszam>309856543</telefonszam>
66. </vendeg>
67.
68. <vendeg ref_szalloda="12345678-2-25" vendeg_szigsz="123456AB">
69.     <nev>
70.
71.         <vendegnev>Kiss Pista</vendegnev>
72.     </nev>
73.     <cim>
74.         <irsz>3515</irsz>
75.         <telepules>Miskolc</telepules>
76.         <utca>Egyetem</utca>
77.         <hsz>1</hsz>
78.     </cim>
79.     <telefonszam>309876543</telefonszam>
80. </vendeg>
81.
82. <vendeg ref_szalloda="12345678-2-25" vendeg_szigsz="654321AB">
83.     <nev>
84.         <vendegnev>Tóth Aladár</vendegnev>
85.     </nev>
86.
87.     <cim>
88.         <irsz>8000</irsz>
89.         <telepules>Székesfehérvár</telepules>
90.         <utca>Erdő utca</utca>
91.         <hsz>140</hsz>
92.     </cim>
93.     <telefonszam>209876543</telefonszam>
94. </vendeg>
95.
96. <alkalmazott alkalmazott_szigsz="135792CA" ref_szalloda="12345678-2-25">
97.     <kor>23</kor>
98.     <szuletesi_datum>
99.         <ido>1997-02-12</ido>
100.        <hely>Kazincbarcika</hely>
101.    </szuletesi_datum>
102.
103.
104.
105.    </alkalmazott>
106.    <alkalmazott alkalmazott_szigsz="246819MA" ref_szalloda="87654321-2-
107.        25">
108.        <kor>38</kor>
109.        <szuletesi_datum>
110.            <ido>1982-12-12</ido>
111.            <hely>Miskolc</hely>
112.        </szuletesi_datum>
113.
114.
115.    </alkalmazott>
116.
117.    <alkalmazott alkalmazott_szigsz="147259GB" ref_szalloda="54654754-2-
118.        25">
119.        <kor>41</kor>
120.        <szuletesi_datum>
121.            <ido>1980-05-05</ido>
122.            <hely>Győr</hely>
123.        </szuletesi_datum>

```

```

124.
125.
126.     </alkalmazott>
127.
128.     <vegzettsegek ref_szigsz="147259GB">
129.         <vegzettseg>Erettsegi</vegzettseg>
130.     </vegzettsegek>
131.
132.     <vegzettsegek ref_szigsz="246819MA">
133.         <vegzettseg>Szakmunkás</vegzettseg>
134.     </vegzettsegek>
135.
136.     <vegzettsegek ref_szigsz="135792CA">
137.         <vegzettseg>Egyetem</vegzettseg>
138.     </vegzettsegek>
139.
140.     <dolgozik ref_szalloda="12345678-2-25" ref_szigsz="147259GB">
141.         <idotartam>8</idotartam>
142.     </dolgozik>
143.
144.     <dolgozik ref_szalloda="87654321-2-25" ref_szigsz="246819MA">
145.         <idotartam>6</idotartam>
146.     </dolgozik>
147.
148.     <dolgozik ref_szalloda="54654754-2-25" ref_szigsz="135792CA">
149.         <idotartam>12</idotartam>
150.     </dolgozik>
151. </Szallodak>

```

1d) Az XML dokumentum alapján XMLSchema készítése:

```

1. <?xml version="1.0" encoding="UTF-8"?>
2.
3.
4. <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
5.
6.     <xs:complexType name="nevTipus">
7.         <xs:sequence>
8.             <xs:element name="vendegnev" type="xs:string"/>
9.
10.        </xs:sequence>
11.    </xs:complexType>
12.    <xs:complexType name="DolgozikTipus">
13.        <xs:sequence>
14.            <xs:element name="idotartam">
15.                <xs:simpleType>
16.                    <xs:restriction base="xs:int">
17.                        <xs:minInclusive value="4"/>
18.                        <xs:maxInclusive value="12"/>
19.                    </xs:restriction>
20.                </xs:simpleType>
21.            </xs:element>
22.        </xs:sequence>
23.        <xs:attribute name="ref_szigsz" use="required">
24.            <xs:simpleType>
25.                <xs:restriction base="xs:string">
26.                    <xs:pattern value="[0-9]{6}[A-Z]{2}"/>
27.                </xs:restriction>

```

```

28.         </xs:simpleType>
29.     </xs:attribute>
30.     <xs:attribute name="ref_szalloda" use="required">
31.         <xs:simpleType>
32.             <xs:restriction base="xs:string">
33.                 <xs:pattern value="[0-9]{8}-[0-9]{1}-[0-9]{2}"/>
34.             </xs:restriction>
35.         </xs:simpleType>
36.     </xs:attribute>
37. </xs:complexType>
38.
39. <xs:complexType name="VegzettsegTipus">
40.     <xs:sequence>
41.         <xs:element name="vegzettseg" type="xs:string"/>
42.
43.     </xs:sequence>
44.     <xs:attribute name="ref_szigsz" use="required">
45.         <xs:simpleType>
46.             <xs:restriction base="xs:string">
47.                 <xs:pattern value="[0-9]{6}[A-Z]{2}"/>
48.             </xs:restriction>
49.         </xs:simpleType>
50.     </xs:attribute>
51. </xs:complexType>
52.
53. <xs:complexType name="cimTipus">
54.     <xs:sequence>
55.         <xs:element name="irsz">
56.             <xs:simpleType>
57.                 <xs:restriction base="xs:int">
58.                     <xs:minInclusive value="1000"/>
59.                     <xs:maxInclusive value="9999"/>
60.                 </xs:restriction>
61.             </xs:simpleType>
62.         </xs:element>
63.         <xs:element name="telepules" type="xs:string"/>
64.         <xs:element name="utca" type="xs:string"/>
65.         <xs:element name="hsz">
66.             <xs:simpleType>
67.                 <xs:restriction base="xs:int">
68.                     <xs:minInclusive value="1"/>
69.                 </xs:restriction>
70.             </xs:simpleType>
71.         </xs:element>
72.     </xs:sequence>
73. </xs:complexType>
74.
75. <xs:complexType name="SzuletesTipus">
76.     <xs:sequence>
77.         <xs:element name="ido">
78.             <xs:simpleType>
79.                 <xs:restriction base="xs:date">
80.                     <xs:pattern value="([12]\d{3}-(0[1-9]|1[0-2])-(0[1-
9]|[12]\d|3[01]))"/>
81.                 </xs:restriction>
82.             </xs:simpleType>
83.         </xs:element>
84.         <xs:element name="hely" type="xs:string"/>
85.
86.
87.     </xs:sequence>
88. </xs:complexType>
89.
90. <xs:simpleType name="telefonszamTipus">
91.     <xs:restriction base="xs:string">
92.         <xs:pattern value="[1-9]{1}[0-9]{8}"/>

```



```

93.         </xs:restriction>
94.     </xs:simpleType>
95.
96.     <xs:simpleType name="arTipus">
97.         <xs:restriction base="xs:int">
98.             <xs:minInclusive value="0"/>
99.         </xs:restriction>
100.     </xs:simpleType>
101.     <xs:simpleType name="korTipus">
102.         <xs:restriction base="xs:int">
103.             <xs:minInclusive value="15"/>
104.             <xs:maxInclusive value="110"/>
105.         </xs:restriction>
106.     </xs:simpleType>
107.
108.     <xs:simpleType name="meretTipus">
109.         <xs:restriction base="xs:int">
110.             <xs:minInclusive value="15"/>
111.             <xs:maxInclusive value="200"/>
112.         </xs:restriction>
113.     </xs:simpleType>
114.
115.     <xs:complexType name="SzallodaTipus" mixed="true">
116.         <xs:sequence>
117.             <xs:element name="nev" type="xs:string"/>
118.             <xs:element name="cim" type="cimTipus"/>
119.             <xs:element name="telefonszam" type="telefonszamTipus"/>
120.
121.         </xs:sequence>
122.         <xs:attribute name="szalloda_adoszam" use="required">
123.             <xs:simpleType>
124.                 <xs:restriction base="xs:string">
125.                     <xs:pattern value="[0-9]{8}-[0-9]{1}-[0-9]{2}"/>
126.                 </xs:restriction>
127.             </xs:simpleType>
128.         </xs:attribute>
129.         <xs:attribute name="ref_szoba" use="required">
130.             <xs:simpleType>
131.                 <xs:restriction base="xs:int">
132.                     <xs:minExclusive value="0"/>
133.                 </xs:restriction>
134.             </xs:simpleType>
135.         </xs:attribute>
136.         <xs:attribute name="ref_vendeg" use="required">
137.             <xs:simpleType>
138.                 <xs:restriction base="xs:string">
139.                     <xs:pattern value="[0-9]{6}[A-Z]{2}"/>
140.                 </xs:restriction>
141.             </xs:simpleType>
142.         </xs:attribute>
143.         <xs:attribute name="ref_alkalmazott" use="required">
144.             <xs:simpleType>
145.                 <xs:restriction base="xs:string">
146.                     <xs:pattern value="[0-9]{6}[A-Z]{2}"/>
147.                 </xs:restriction>
148.             </xs:simpleType>
149.         </xs:attribute>
150.     </xs:complexType>
151.
152.     <xs:complexType name="VendegTipus" mixed="true">
153.         <xs:sequence>
154.             <xs:element name="nev" type="nevTipus"/>
155.             <xs:element name="cim" type="cimTipus"/>
156.             <xs:element name="telefonszam" type="telefonszamTipus"/>
157.         </xs:sequence>
158.         <xs:attribute name="vendeg_szigsz" use="required">

```

```

159.         <xs:simpleType>
160.             <xs:restriction base="xs:string">
161.                 <xs:pattern value="[0-9]{6}[A-Z]{2}" />
162.             </xs:restriction>
163.         </xs:simpleType>
164.     </xs:attribute>
165.     <xs:attribute name="ref_szalloda" use="required">
166.         <xs:simpleType>
167.             <xs:restriction base="xs:string">
168.                 <xs:pattern value="[0-9]{8}-[0-9]{1}-[0-9]{2}" />
169.             </xs:restriction>
170.         </xs:simpleType>
171.     </xs:attribute>
172. </xs:complexType>
173.
174. <xs:complexType name="AlkalmazottTipus" mixed="true">
175.     <xs:sequence>
176.         <xs:element name="kor" type="korTipus" />
177.         <xs:element name="szuletesi_datum" type="SzuletesiTipus" />
178.     </xs:sequence>
179.     <xs:attribute name="alkalmazott_szigsz" use="required">
180.         <xs:simpleType>
181.             <xs:restriction base="xs:string">
182.                 <xs:pattern value="[0-9]{6}[A-Z]{2}" />
183.             </xs:restriction>
184.         </xs:simpleType>
185.     </xs:attribute>
186.     <xs:attribute name="ref_szalloda" use="required">
187.         <xs:simpleType>
188.             <xs:restriction base="xs:string">
189.                 <xs:pattern value="[0-9]{8}-[0-9]{1}-[0-9]{2}" />
190.             </xs:restriction>
191.         </xs:simpleType>
192.     </xs:attribute>
193. </xs:complexType>
194.
195. <xs:complexType name="SzobaTipus" mixed="true">
196.     <xs:sequence>
197.         <xs:element name="meret" type="meretTipus" />
198.         <xs:element name="ar" type="arTipus" />
199.         <xs:element name="tipus" type="xs:string" />
200.     </xs:sequence>
201.     <xs:attribute name="szoba_ajtoszam" use="required">
202.         <xs:simpleType>
203.             <xs:restriction base="xs:int">
204.                 <xs:minExclusive value="0" />
205.             </xs:restriction>
206.         </xs:simpleType>
207.     </xs:attribute>
208.     <xs:attribute name="ref_szalloda" use="required">
209.         <xs:simpleType>
210.             <xs:restriction base="xs:string">
211.                 <xs:pattern value="[0-9]{8}-[0-9]{1}-[0-9]{2}" />
212.             </xs:restriction>
213.         </xs:simpleType>
214.     </xs:attribute>
215. </xs:complexType>
216.
217. <xs:element name="Szallodak">
218.     <xs:complexType>
219.         <xs:sequence>
220.             <xs:element name="szalloda" type="SzallodaTipus" maxOccurs="unbounded" />
221.             <xs:element name="szoba" type="SzobaTipus" maxOccurs="unbounded" />

```

```

222.             <xs:element name="vendeg" type="VendegTipus" maxOccurs="unbou
nded"/>
223.             <xs:element name="alkalmazott" type="AlkalmazottTipus" maxOcc
urs="unbounded"/>
224.             <xs:element name="vegzettsegek" type = "VegzettsegTipus" maxO
ccurs="unbounded"/>
225.             <xs:element name="dolgozik" type = "DolgozikTipus" maxOccurs=
"unbounded"/>
226.         </xs:sequence>
227.     </xs:complexType>
228.     <xs:key name="szallodaKey">
229.         <xs:selector xpath="szalloda"/>
230.         <xs:field xpath="@szalloda_adoszam"/>
231.     </xs:key>
232.     <xs:key name="szobaKey">
233.         <xs:selector xpath="szoba"/>
234.         <xs:field xpath="@szoba_ajtoszam"/>
235.     </xs:key>
236.     <xs:key name="vendegKey">
237.         <xs:selector xpath="vendeg"/>
238.         <xs:field xpath="@vendeg_szigsz"/>
239.     </xs:key>
240.
241.     <xs:key name="alkalmazottKey">
242.         <xs:selector xpath="alkalmazott"/>
243.         <xs:field xpath="@alkalmazott_szigsz"/>
244.     </xs:key>
245.
246.     <xs:keyref refer="vendegKey" name="refVendeg_sz">
247.         <xs:selector xpath="./szalloda"></xs:selector>
248.         <xs:field xpath="@ref_vendeg"></xs:field>
249.     </xs:keyref>
250.
251.     <xs:keyref refer="alkalmazottKey" name="refAlkalmazott_sz">
252.         <xs:selector xpath="./szalloda"></xs:selector>
253.         <xs:field xpath="@ref_alkalmazott"></xs:field>
254.     </xs:keyref>
255.
256.
257.     <xs:keyref refer="szobaKey" name="refSzalloda">
258.         <xs:selector xpath="./szalloda"/>
259.         <xs:field xpath="@ref_szoba"/>
260.     </xs:keyref>
261.
262.     <xs:keyref refer="szallodaKey" name="refSzoba">
263.         <xs:selector xpath="./szoba"/>
264.         <xs:field xpath="@ref_szalloda"/>
265.     </xs:keyref>
266.     <xs:keyref refer="szallodaKey" name="refVendeg">
267.         <xs:selector xpath="./vendeg"/>
268.         <xs:field xpath="@ref_szalloda"/>
269.     </xs:keyref>
270.     <xs:keyref refer="szallodaKey" name="refAlkalmazott">
271.         <xs:selector xpath="./alkalmazott"/>
272.         <xs:field xpath="@ref_szalloda"/>
273.     </xs:keyref>
274.     <xs:keyref refer="szallodaKey" name="refDolgozik">
275.         <xs:selector xpath="./dolgozik"/>
276.         <xs:field xpath="@ref_szalloda"/>
277.     </xs:keyref>
278.
279.     <xs:keyref refer="alkalmazottKey" name="refVegzettsegek_sz">
280.         <xs:selector xpath="./vegzettsegek"></xs:selector>
281.         <xs:field xpath="@ref_alkalmazott"></xs:field>
282.     </xs:keyref>
283.     <xs:keyref refer="alkalmazottKey" name="refDolgozik_sz">

```

```

284.         <xs:selector xpath="./dolgozik"></xs:selector>
285.         <xs:field xpath="@ref_alkalmazott"></xs:field>
286.     </xs:keyref>
287.
288. </xs:element>
289.
290.
291. </xs:schema>

```

2. feladat

A feladat egy DOM program készítése az XML dokumentum adatainak adminisztrálása alapján:

2a) adatolvasás

```

1. package hu.domparse.do425g;
2.
3. import java.io.File;
4. import java.io.IOException;
5.
6. import javax.xml.parsers.*;
7.
8. import org.w3c.dom.*;
9.
10. public class DOMReadD0425G {
11.
12.     public static void main(String[] args) {
13.         NodeList nodelist;
14.         try {
15.             DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
16.             DocumentBuilder builder = factory.newDocumentBuilder();
17.             //fájl beolvasása
18.             Document document = builder.parse(new File("XMLD0425G.xml"));
19.             document.getDocumentElement().normalize();
20.             //Gyökér elem keresése
21.             System.out.println("Root element:" + document.getDocumentElement().getNo
deName());
22.             System.out.println("_____");
23.             //Aktuális elem meghatározása
24.             nodelist = document.getElementsByTagName("szalloda");
25.
26.             for (int i = 0; i < nodelist.getLength(); i++) {
27.                 Node node = nodelist.item(i);
28.                 System.out.println("\nAktuális elem: " + node.getNodeName());
29.                 //Szálloda adatainak kiírása
30.                 if (node.getNodeType() == Node.ELEMENT_NODE) {
31.                     Element element = (Element) node;
32.                     System.out.println("Szálloda adószáma: " + element.getAttribute(
"szalloda_adoszam"));
33.                     System.out.println("Szállodához tartozó szobaszám: " + element.g
etAttribute("ref_szoba"));
34.                     System.out.println("Szállodához tartozó vendég személyigazolvány
szám: " + element.getAttribute("ref_vendeg"));
35.                     System.out.println("Szálloda Alkalmazott személyigazolvány száma
: " + element.getAttribute("ref_alkalmazott"));
36.                     System.out
37.                         .println("Szálloda Neve: " + element.getElementsByTagName
e("nev").item(0).getTextContent());
38.                     //Szállodán belüli cím kiírása

```

```

39.         Node nodecim = nodelist.item(i);
40.         if (nodecim.getNodeType() == Node.ELEMENT_NODE) {
41.             Element elementcim = (Element) node;
42.             System.out.println(
43.                 "Írányítószám: " + element.getElementsByTagName("irs
z").item(0).getTextContent());
44.             System.out.println(
45.                 "Település: " + element.getElementsByTagName("telepu
les").item(0).getTextContent());
46.             System.out.println("Utca: " + element.getElementsByTagName("
utca").item(0).getTextContent());
47.             System.out.println("Házszám: " + element.getElementsByTagName("
hsz").item(0).getTextContent());
48.         }
49.     }
50. }
51. //Aktuális elem meghatározása
52. nodelist = document.getElementsByTagName("szoba");
53. for(int i = 0; i < nodelist.getLength(); i++) {
54.     Node node = nodelist.item(i);
55.     System.out.println("\nAktuális elem: " + node.getNodeName());
56.     //Szoba adatainak kiírása
57.     if (node.getNodeType() == Node.ELEMENT_NODE) {
58.         Element element = (Element) node;
59.         System.out.println("Szoba ajtószáma: " + element.getAttribute("s
zoba_ajtoszam"));
60.         System.out.println("Szálloda adószáma: " + element.getAttribute(
"ref_szalloda"));
61.         System.out.println("Szoba mérete: " + element.getElementsByTagName("
meret").item(0).getTextContent());
62.         System.out.println("Szoba ára: " + element.getElementsByTagName("
ar").item(0).getTextContent());
63.         System.out.println("Szoba típusa: " + element.getElementsByTagName("
tipus").item(0).getTextContent());
64.     }
65. }
66. }
67. //Aktuális elem meghatározása
68. nodelist = document.getElementsByTagName("vendeg");
69. for(int i = 0; i < nodelist.getLength(); i++) {
70.     Node node = nodelist.item(i);
71.     System.out.println("\nAktuális elem: " + node.getNodeName());
72.     //Vendég adatainak kiírása
73.     if (node.getNodeType() == Node.ELEMENT_NODE) {
74.         Element element = (Element) node;
75.         System.out.println("Vendég Személyigazolvány száma: " + element.
getAttribute("vendeg_szigsz"));
76.         System.out.println("Szálloda adószáma: " + element.getAttribute(
"ref_szalloda"));
77.         System.out.println("Vendég Neve: " + element.getElementsByTagName("
vendegnev").item(0).getTextContent());
78.         Node nodecim = nodelist.item(i);
79.         //Vendégeken belüli címek kiírása
80.         if (nodecim.getNodeType() == Node.ELEMENT_NODE) {
81.             Element elementcim = (Element) node;
82.             System.out.println(
83.                 "Írányítószám: " + element.getElementsByTagName("irs
z").item(0).getTextContent());
84.             System.out.println(
85.                 "Település: " + element.getElementsByTagName("telepu
les").item(0).getTextContent());
86.             System.out.println("Utca: " + element.getElementsByTagName("
utca").item(0).getTextContent());
87.             System.out.println("Házszám: " + element.getElementsByTagName("
hsz").item(0).getTextContent());
88.         }

```

```

89.         System.out.println("Vendég telefonszáma: " + element.getElements
ByTagName("telefonszam").item(0).getTextContent());
90.
91.     }
92. }
93. //Aktuális elem meghatározása
94. nodelist = document.getElementsByTagName("alkalmazott");
95. for(int i = 0; i < nodelist.getLength(); i++) {
96.     Node node = nodelist.item(i);
97.     System.out.println("\nAktuális elem: " + node.getNodeName());
98.     //Alkalmazott adatainak kiírása
99.     if(node.getNodeType() == Node.ELEMENT_NODE) {
100.         Element element = (Element) node;
101.         System.out.println("Alkalmazott Személyigazolvány száma:
" + element.getAttribute("alkalmazott_szigsz"));
102.         System.out.println("Szálloda száma: " + element.getAttrib
ute("ref_szalloda"));
103.         System.out.println("Alkalmazott kora: " + element.getElem
entsByTagName("kor").item(0).getTextContent());
104.         //Alkalmazotton belüli születési adatok kiírása
105.         Node nodeszulesdatum = nodelist.item(i);
106.         if (nodeszulesdatum.getNodeType() == Node.ELEMENT_NODE)
        {
107.             Element elementcim = (Element) node;
108.             System.out.println("Születési idő: " + element.getEle
mentsByTagName("ido").item(0).getTextContent());
109.             System.out.println(
110.                 "Születési hely: " + element.getElementsByTag
Name("hely").item(0).getTextContent());
111.
112.         }
113.     }
114. }
115. //Aktuális elem meghatározása
116. nodelist = document.getElementsByTagName("vegzettsegek");
117. for(int i = 0; i < nodelist.getLength(); i++) {
118.     Node node = nodelist.item(i);
119.     System.out.println("\nAktuális elem: " + node.getNodeName());
120.
121.     //Vegzettségek adatainak kiírása
122.     if(node.getNodeType() == Node.ELEMENT_NODE) {
123.         Element element = (Element) node;
124.         System.out.println("Végzettséggel rendelkező Személyigazo
lvány száma: " + element.getAttribute("ref_szigsz"));
125.         System.out.println("Végzettség: " + element.getElementsBy
TagName("vegzettseg").item(0).getTextContent());
126.     }
127. }
128. //Aktuális elem meghatározása
129. nodelist = document.getElementsByTagName("dolgozik");
130. for(int i = 0; i < nodelist.getLength(); i++) {
131.     Node node = nodelist.item(i);
132.     System.out.println("\nAktuális elem: " + node.getNodeName());
133.
134.     //Dolgozik adatainak kiírása
135.     if(node.getNodeType() == Node.ELEMENT_NODE) {
136.         Element element = (Element) node;
137.         System.out.println("Dolgozói Személyigazolvány szám: " +
element.getAttribute("ref_szigsz"));
138.         System.out.println("Dolgozói Szálloda száma: " + element.
getAttribute("ref_szalloda"));
139.         System.out.println("Munkavégzés időtartama: " + element.g
etElementsByTagName("idotartam").item(0).getTextContent());
140.     }
141. }

```

```

141.         } catch (ParserConfigurationException pce) {
142.             pce.printStackTrace();
143.         } catch (Exception e) {
144.             e.printStackTrace();
145.         }
146.
147.     }
148.
149. }

```

2b) adatmódosítás

```

1. package hu.domparse.do425g;
2.
3. import java.io.File;
4.
5. import javax.xml.parsers.DocumentBuilder;
6. import javax.xml.parsers.DocumentBuilderFactory;
7. import javax.xml.parsers.ParserConfigurationException;
8. import javax.xml.transform.Result;
9. import javax.xml.transform.Source;
10. import javax.xml.transform.Transformer;
11. import javax.xml.transform.TransformerFactory;
12. import javax.xml.transform.dom.DOMSource;
13. import javax.xml.transform.stream.StreamResult;
14.
15. import org.w3c.dom.*;
16.
17.
18. public class DOMModifyD0425G {
19.
20.     public static void main(String[] args) {
21.         NodeList nodelist;
22.         try {
23.             DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
24.             DocumentBuilder builder = factory.newDocumentBuilder();
25.             // Fájlt beolvasása
26.             Document document = builder.parse(new File("XMLD0425G.xml"));
27.             document.getDocumentElement().normalize();
28.             // Aktuális elem meghatározása
29.             nodelist = document.getElementsByTagName("szalloda");
30.             for(int i = 0; i < nodelist.getLength(); i++) {
31.                 Node node = nodelist.item(i);
32.                 System.out.println("\nAktuális elem:" + node.getNodeName());
33.                 //Szálloda adatainak kiírása
34.                 if(node.getNodeType() == Node.ELEMENT_NODE) {
35.                     Element element = (Element) node;
36.
37.
38.                     System.out.println("Szálloda adószáma:" + element.getAttribute("s
zalloda_adoszam"));
39.                     System.out.println("Szálloda szobaszám:" + element.getAttribute("
ref_szoba"));
40.                     System.out.println("Szálloda vendég személyigazolvány száma:" + e
lement.getAttribute("ref_vendeg"));
41.                     System.out.println("Szálloda alkalmazott személyigazolvány száma
:" + element.getAttribute("ref_alkalmazott"));
42.                     Node nodecim = nodelist.item(i);
43.                     //Szálloda belüli lakcím adatok kiírása
44.                     if(nodecim.getNodeType() == Node.ELEMENT_NODE) {
45.                         Element elementcim = (Element) node;
46.                         //Szálloda adószám vizsgálata, ha egyezik akkor az adatok mó
dosítása

```

```

47.         if(element.getAttribute("szalloda_adoszam").equals("12345678
-2-25")) {
48.             elementcim.getElementsByTagName("irsz").item(0).setText
Content("3700");
49.             elementcim.getElementsByTagName("telepules").item(0).se
tTextContent("Kazincbarcika");
50.             elementcim.getElementsByTagName("utca").item(0).setTextC
ontent("Építők utca");
51.             elementcim.getElementsByTagName("hsz").item(0).setTextC
ontent("44");
52.         }
53.         System.out.println(
54.             "Irányítószám: " + element.getElementsByTagName(
"irsz").item(0).getTextContent());
55.         System.out.println(
56.             "Település: " + element.getElementsByTagName("te
lepules").item(0).getTextContent());
57.         System.out.println("Utca: " + element.getElementsByTagName("utca").item(0).getTextContent());
58.         System.out.println("Házszám: " + element.getElementsByTagName("hsz").item(0).getTextContent());
59.     }
60. }
61. }
62. //módosított xml fájl létrehozása
63. Transformer transformer = TransformerFactory.newInstance().newTransformer();
64. Source input = new DOMSource(document);
65. Result output = new StreamResult(new File("XMLD0425GModify.xml"));
66. transformer.transform(input, output);
67. }catch(ParserConfigurationException pce) {
68.     pce.printStackTrace();
69. } catch(Exception e) {
70.     e.printStackTrace();
71. }
72.
73. }
74.
75. }

```