

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Autó kölcsönző

Készítette: **Pázmán András**

Neptunkód: **H2Z4X3**

Dátum: 2023.11.29

Tartalomjegyzék

A feladat leírása	3
1. feladat	4
1a) Az adatbázis ER modell tervezése	4
1b) Az adatbázis konvertálása XDM modellre	5
1c) Az XDM modell alapján XML dokumentum készítése:	6
1d) Az XML dokumentum alapján XMLSchema készítése	10
2. feladat	14
2a) adatolvasás	14
2b) adatmódosítás	19
2c) adatlekérdezés	23
2d) adatírás	23

A feladat leírása:

A feladat egy autókölcsönző vállalkozáshoz kapcsolódik, amely több telephellyel rendelkezik országszerte. A telephelyeken különböző autók találhatóak amelyeket az ügyfelek kibérelhetnek.

A bérlet adminisztrálását és a telep működését az alkalmazottak biztosítják akik különböző pozíciókban dolgoznak. A telephely tulajdonosához tartozik az adott telep melynek felügyeli a működését. Egy tulajdonoshoz egy telep tartozik, ő kezeli a telep bevételeit. Ezt jelzi a bevétel tulajdonság. Minden telephelynek más a kapacitása mivel az autókat tárolni kell ezért a fennálló kapacitás korlátozott. Tovább a telep rendelkezik egy minősítési számmal (beosztás) melynek értéke A, B, C, D, E szerint oszlik meg.

Az A kategória a legjobb értékelés, az E a legrosszabb. Az értékelést sok minden befolyásolhatja, akár a vevői értékelés, telep kapacitás stb.

Ha egy személy egy autót szeretne kibérelni akkor az adatait az adatbázisban rögzítjük. Ennek több oka is van, az adatok alapján könnyebb visszakeresni az adott személy ha esetlegesen kár keletkezik az autóba. Egy alkalmazott egy időben csak egy autót bérelhet ki, ezáltal egy adott autó is csak 1 ügyfélnél lehet ugyan abban az időpontban. Ezzel kizárjuk azt hogy többen is ugyan azt az autót foglalják re ugyan arra az időpontra.

Az alkalmazottak pozíciójuknak megfelelően szerezhethetnek plusz képzéseket melyekkel növelhetik a fizetésüket. A különböző juttatások 10 000- 20 000- 30 000 Ft összegekben oszlanak meg. A képzéshez különböző szint és nyelv is tartozik melyek tovább befolyásolják a juttatás mértékét. Egy alkalmazott több képzéssel is rendelkezhet. A képzések közé beletartozik a nyelvvizsga bizonyítvány is is.

1. feladat

1a) Az adatbázis ER modell tervezése

Az alábbi ER modell tartalmazza az autókölcsönző adatmodell felépítését.

A ER modellben 6 egyed található melyek között megtalálható 1:1, 1:N, illetve N:M kapcsolat is.

Az egyedek minimum 4 tulajdonsággal rendelkeznek, ezeken kívül egyedi azonosításként kaptak egy ID-t.

Tulajdonos:

- Név (Tulajdonos neve)
- Adószám (Tulajdonos adószáma)
- Bevétel (A telep bevétele)
- TelSzám (Tulajdonos telefonszáma)
- TulID (primary key)

Telep:

- Kapacitás (a tárolható autók száma)
- Cím (összetett tulajdonság, tartalmazza a települést, utca, hászámot)
- Beosztás (A telep minősítését, kategóriáját tartalmazza (A, B, C, D vagy E)
- TelepID (primary key)

Autók:

- Modell (az autó típusát tartalmazza)
- Rendszám (a bérelhető autó rendszáma)
- Alvázszám (az autó alvázszáma)
- Gyártási év (az autó gyártási éve)
- AutoID (primary key)

Bérlő:

- Név (A bérlő neve)
- TelSzám (A bérlő telefon száma)
- Email (A bérlő email címe)
- Jogosítvány szám (Tulajdonos telefonszáma)
- BérlőID (primary key)

Alkalmazottak:

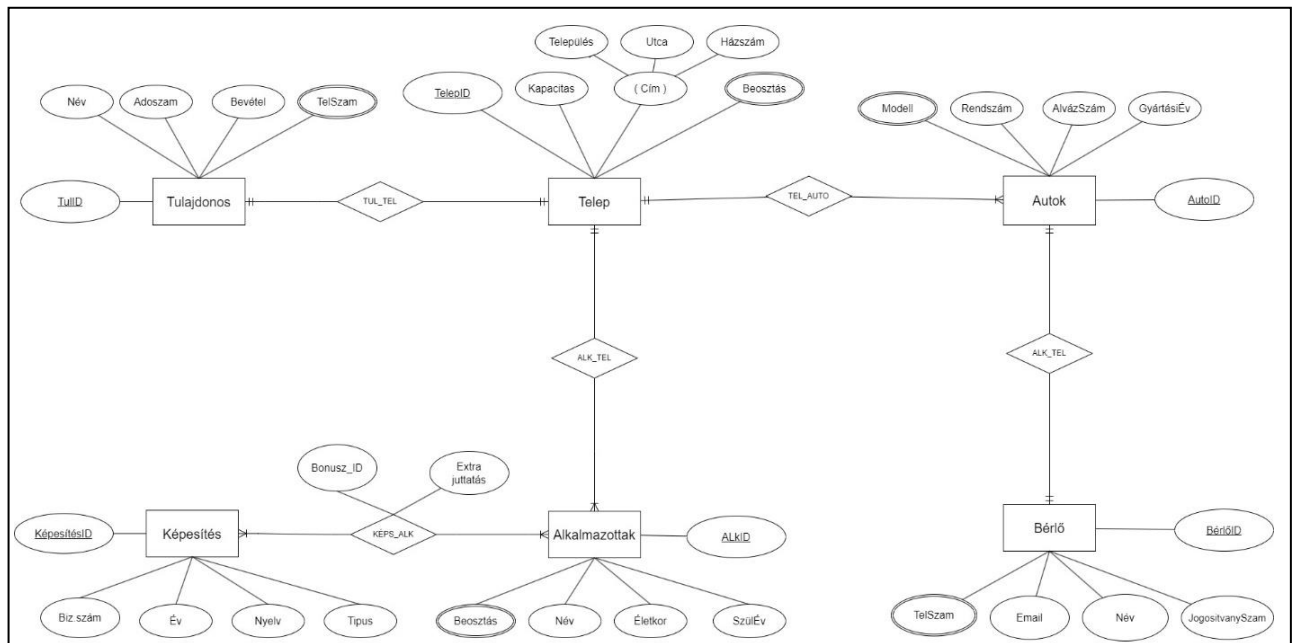
- Név (alkalmazott neve)
- Beosztás (a feladatkör amit az alkalmazott ellát)
- SzülÉv (Az alkalmazott születési éve)
- Életkor (Az alkalmazott életkora)
- AlkID (primary key)

Képesítés:

- Típus (a bizonyítvány típusa)
- Nyelv
- Év
- BizSzám (bizonyítvány száma)
- KépesítésID (primary key)

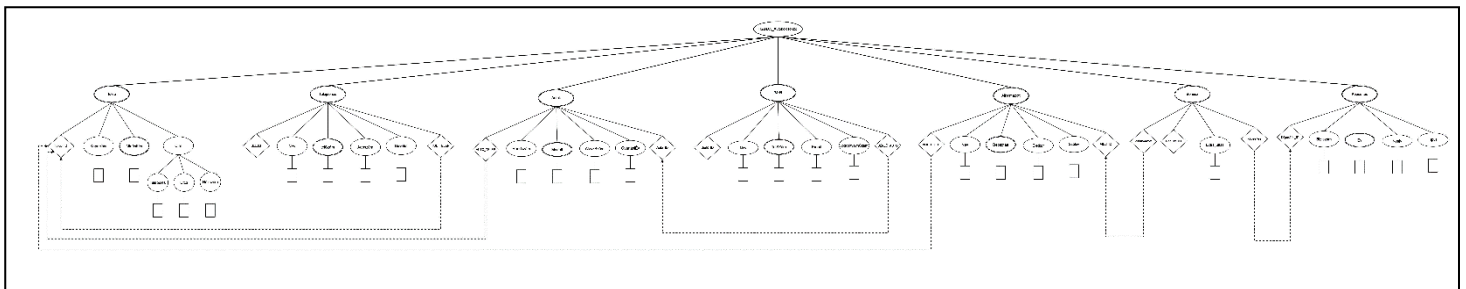
Bonusz:

- Extra (a juttatás mértéke)
- BonuszID (primary key)



1b) Az adatbázis konvertálása XDM modellre

Az XDM modell elkészítése során kialakításra kerültek a kapcsolatok az elsődleges és idegen kulcsok között. Továbbá a modell kivitelezése alatt ügyelni kellett hogy a kapcsolatokat jelölő szaggatott vonalak ne keresztezzék egymást mivel ez rontja a modell átláthatóságát.



1c) Az XDM modell alapján XML dokumentum készítése:

Az XDM modell átkonvertálva XML modellre az elsődleges és másodlagos kulcsokból attribútumok lettek. Minden elem legalább 3 példányban kerül átkonvertálásra.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2
3
4  <H2Z4X3_autokolcsonzo xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
5  xs:noNamespaceSchemaLocation="XMLSchemah2z4x3.xsd">
6
7      <!-- TELEP -->
8      <telep TELEP_ID = "1">
9          <kapacitás>5</kapacitás>
10         <cím>
11             <település>Budapest</település>
12             <utca>Benczúr_utca</utca>
13             <házszám>23</házszám>
14         </cím>
15         <minősítés>B</minősítés>
16     </telep>
17
18     <telep TELEP_ID = "2">
19         <kapacitás>8</kapacitás>
20         <cím>
21             <település>Budapest</település>
22             <utca>Jókai_utca</utca>
23             <házszám>14</házszám>
24         </cím>
25         <minősítés>A</minősítés>
26     </telep>
27
28     <telep TELEP_ID = "3">
29         <kapacitás>6</kapacitás>
30         <cím>
31             <település>Szeged</település>
32             <utca>Szabadság_utca </utca>
33             <házszám>23</házszám>
34         </cím>
35         <minősítés>C</minősítés>
36     </telep>
37
38     <telep TELEP_ID = "4">
39         <kapacitás>7</kapacitás>
40         <cím>
41             <település>Siófok</település>
42             <utca>Balaton_utca</utca>
43             <házszám>45</házszám>
44         </cím>
45         <minősítés>A</minősítés>
46     </telep>
```

```

49 <!-- TULAJDONOS-->
50 <tulajdonos TUL_ID="1" TUL_TELEP="1" >
51   <név>Kis János</név>
52   <telszám>067022344555</telszám>
53   <adoszam>12345678-1-11</adoszam>
54   <bevétel>1500000</bevétel>
55 </tulajdonos>
56
57 <tulajdonos TUL_ID="2" TUL_TELEP="2" >
58   <név>Cicz Imre</név>
59   <telszám>06205678899</telszám>
60   <adoszam>18384763-2-24</adoszam>
61   <bevétel>1234560</bevétel>
62 </tulajdonos>
63
64 <tulajdonos TUL_ID="3" TUL_TELEP="3" >
65   <név>Cset Elek</név>
66   <telszám>06703452233</telszám>
67   <adoszam>12345679-3-13</adoszam>
68   <bevétel>1800000</bevétel>
69 </tulajdonos>
70
71 <tulajdonos TUL_ID="4" TUL_TELEP="4" >
72   <név>Kis Anikó</név>
73   <telszám>06304567788</telszám>
74   <adoszam>12341236-2-31</adoszam>
75   <bevétel>1850000</bevétel>
76 </tulajdonos>
77
78
79 <!-- AUTOK-->
80 <autok AUTO_ID="1" AUTO_TELEP="1" >
81   <rendszám>PSY-213</rendszám>
82   <modell>Jaguar_F-TYPE</modell>
83   <alvázzszám>JZMMA18P200411817</alvázzszám>
84   <gyártásiév>2019</gyártásiév>
85 </autok>
86
87 <autok AUTO_ID="2" AUTO_TELEP="2" >
88   <rendszám>RYZ-022</rendszám>
89   <modell>Mercedes_S63_AMG</modell>
90   <alvázzszám>JZMMA18P201111111</alvázzszám>
91   <gyártásiév>2020</gyártásiév>
92 </autok>
93
94 <autok AUTO_ID="3" AUTO_TELEP="3" >
95   <rendszám>SUV-123</rendszám>
96   <modell>Volvo_XC-90</modell>
97   <alvázzszám>KMMMA18P202345234</alvázzszám>
98   <gyártásiév>2021</gyártásiév>
99 </autok>
100
101 <autok AUTO_ID="4" AUTO_TELEP="2" >
102   <rendszám>SRY-345</rendszám>
103   <modell>Audi_A5</modell>
104   <alvázzszám>PZMMA18P200434567</alvázzszám>
105   <gyártásiév>2020</gyártásiév>
106 </autok>
107
108 <autok AUTO_ID="5" AUTO_TELEP="4" >
109   <rendszám>PSY-234</rendszám>
110   <modell>BMW_M5_Competition</modell>
111   <alvázzszám>KMMMA18P200413259</alvázzszám>
112   <gyártásiév>2018</gyártásiév>
113 </autok>

```

```

117 <!-- BÉRLŐ-->
118 <bérlő BÉRLŐ_ID="1" BÉRLŐ_AUTO="2">
119   <név>Diz Elek</név>
120   <telszám>06703439988</telszám>
121   <email>dizelek@gmail.com</email>
122   <jogosítványszám>CM001001</jogosítványszám>
123 </bérlő>
124
125 <bérlő BÉRLŐ_ID="2" BÉRLŐ_AUTO="3">
126   <név>Farkas Piroska</név>
127   <telszám>06209876677</telszám>
128   <email>piroksa@gfreemail.hu</email>
129   <jogosítványszám>WW001002</jogosítványszám>
130 </bérlő>
131
132 <bérlő BÉRLŐ_ID="3" BÉRLŐ_AUTO="4">
133   <név>Feles Elek</név>
134   <telszám>06708995544</telszám>
135   <email>felelek@gmail.com</email>
136   <jogosítványszám>MM020105</jogosítványszám>
137 </bérlő>
138
139 <bérlő BÉRLŐ_ID="4" BÉRLŐ_AUTO="5">
140   <név>Git Áron</név>
141   <telszám>06204563322</telszám>
142   <email>gitaron@citromail.hu</email>
143   <jogosítványszám>KK011501</jogosítványszám>
144 </bérlő>
145
146 <!-- ALKALMAZOTTAK-->
147 <alkalmazott ALK_ID="1" ALK_TELEP="2">
148   <név>Hideg Ottó</név>
149   <beosztás>Adminisztrátor</beosztás>
150   <életkor>48</életkor>
151   <születésiév>1975</születésiév>
152 </alkalmazott>
153
154 <alkalmazott ALK_ID="2" ALK_TELEP="3">
155   <név>Kis Eszter</név>
156   <beosztás>Üzletvezető</beosztás>
157   <életkor>45</életkor>
158   <születésiév>1978</születésiév>
159 </alkalmazott>
160
161 <alkalmazott ALK_ID="3" ALK_TELEP="1">
162   <név>Pat Tamás</név>
163   <beosztás>supervisor</beosztás>
164   <életkor>35</életkor>
165   <születésiév>1988</születésiév>
166 </alkalmazott>
167
168 <alkalmazott ALK_ID="4" ALK_TELEP="4">
169   <név>Tra Pista</név>
170   <beosztás>marketing_manager</beosztás>
171   <életkor>39</életkor>
172   <születésiév>1984</születésiév>
173 </alkalmazott>
174
175 <alkalmazott ALK_ID="5" ALK_TELEP="2">
176   <név>Viz Elma</név>
177   <beosztás>hr_vezető</beosztás>
178   <életkor>45</életkor>
179   <születésiév>1978</születésiév>
180 </alkalmazott>
181

```



```
183 <!-- BONUSZ-->
184 <bonusz BONUSZ_ID="1" ALKALMAZOTT_FK = "1" KÉPESÍTÉS_FK="1" >
185 |   <extra>10000</extra>
186 </bonusz>
187
188 <bonusz BONUSZ_ID="2" ALKALMAZOTT_FK = "2" KÉPESÍTÉS_FK="2" >
189 |   <extra>25000</extra>
190 </bonusz>
191
192 <bonusz BONUSZ_ID="3" ALKALMAZOTT_FK = "3" KÉPESÍTÉS_FK="3" >
193 |   <extra>30000</extra>
194 </bonusz>
195
196 <!-- Képesítés-->
197 <képesítés KÉPESÍTÉS_ID="1">
198 |   <bizosítványszám>7245889</bizosítványszám>
199 |   <év>2021</év>
200 |   <nyelv>angol</nyelv>
201 |   <típus>A2</típus>
202 </képesítés>
203
204 <képesítés KÉPESÍTÉS_ID="2">
205 |   <bizosítványszám>7245867</bizosítványszám>
206 |   <év>2015</év>
207 |   <nyelv>magyar</nyelv>
208 |   <típus>B2</típus>
209 </képesítés>
210
211 <képesítés KÉPESÍTÉS_ID="3">
212 |   <bizosítványszám>7224589</bizosítványszám>
213 |   <év>2010</év>
214 |   <nyelv>angol</nyelv>
215 |   <típus>C1</típus>
216 </képesítés>
217
218 <képesítés KÉPESÍTÉS_ID="4">
219 |   <bizosítványszám>7298174</bizosítványszám>
220 |   <év>2011</év>
221 |   <nyelv>angol</nyelv>
222 |   <típus>B1</típus>
223 </képesítés>
224
225 </H2Z4X3_autokolcsonzo>
```

1d) Az XML dokumentum alapján XMLSchema készítése

- saját típusok, ref, key, keyref, speciális elemek.

XML dokumentumhoz elkészítettem az XML Schemat. Elsődlegesen az egyszerű és saját típusokat gyűjtöttem ki, ezután komplex típusok, majd elsődleges kulcsok, idegen kulcsok.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
4
5      <!-- Egyszerű típusok kigyűjtése, saját típusok meghatározása -->
6      <xs:element name="kapacitás" type="xs:integer" />
7      <xs:element name="minősítés" type="minősítésTípus" />
8      <xs:element name="név" type="xs:string" />
9      <xs:element name="cím" type="xs:string" />
10     <xs:element name="telszám" type="xs:string" />
11     <xs:element name="adoszam" type="adószámTípus" />
12     <xs:element name="bevétel" type="xs:integer" />
13     <xs:element name="rendszám" type="rendszámTípus" />
14     <xs:element name="modell" type="xs:string" />
15     <xs:element name="alvázszám" type="alvázszámTípus" />
16     <xs:element name="gyártásiév" type="xs:gYear" />
17     <xs:element name="email" type="xs:string" />
18     <xs:element name="jogosítványszam" type="jogosítványszámTípus" />
19     <xs:element name="beosztás" type="xs:string" />
20     <xs:element name="életkor" type="xs:integer" />
21     <xs:element name="születésiév" type="xs:gYear" />
22     <xs:element name="extra" type="extraTípus" />
23     <xs:element name="bizosítványszám" type="bizosítványszámTípus" />
24     <xs:element name="év" type="xs:gYear" />
25     <xs:element name="nyelv" type="xs:string" />
26     <xs:element name="típus" type="típusTípus" />
27
28     <xs:simpleType name="minősítésTípus">
29         <xs:restriction base="xs:string">
30             <xs:enumeration value="A" />
31             <xs:enumeration value="B" />
32             <xs:enumeration value="C" />
33             <xs:enumeration value="D" />
34             <xs:enumeration value="E" />
35         </xs:restriction>
36     </xs:simpleType>
37     <xs:simpleType name="adószámTípus">
38         <xs:restriction base="xs:string">
39             <xs:pattern value="(\d{8})-(\d)-(\d{2})" />
40         </xs:restriction>
41     </xs:simpleType>
42     <xs:simpleType name="bevételTípus">
43         <xs:restriction base="xs:int">
44             <xs:minInclusive value="100000" />
45             <xs:maxInclusive value="10000000" />
46         </xs:restriction>
47     </xs:simpleType>
```

```

48 <xs:simpleType name="rendszámTípus">
49 |   <xs:restriction base="xs:string">
50 |     <xs:pattern value="([A-Z]{3}-\d{3})" />
51 |   </xs:restriction>
52 </xs:simpleType>
53 <xs:simpleType name="alvázszámTípus">
54 |   <xs:restriction base="xs:string">
55 |     <xs:length value="17" />
56 |   </xs:restriction>
57 </xs:simpleType>
58 <xs:simpleType name="jogosítványszámTípus">
59 |   <xs:restriction base="xs:string">
60 |     <xs:pattern value="([A-Z]{2}\d{6})" />
61 |   </xs:restriction>
62 </xs:simpleType>
63 <xs:simpleType name="extraTípus">
64 |   <xs:restriction base="xs:int">
65 |     <xs:minInclusive value="10000" />
66 |     <xs:maxInclusive value="50000" />
67 |   </xs:restriction>
68 </xs:simpleType>
69 <xs:simpleType name="tipusTípus">
70 |   <xs:restriction base="xs:string">
71 |     <xs:enumeration value="A1" />
72 |     <xs:enumeration value="A2" />
73 |     <xs:enumeration value="B1" />
74 |     <xs:enumeration value="B2" />
75 |     <xs:enumeration value="C1" />
76 |     <xs:enumeration value="C2" />
77 |   </xs:restriction>
78 </xs:simpleType>
79 <xs:simpleType name="bizosítványszámTípus">
80 |   <xs:restriction base="xs:string">
81 |     <xs:length value="7" />
82 |   </xs:restriction>
83 </xs:simpleType>
84
85 <!-- Komplex típushoz saját típus-->
86 <xs:complexType name="telepTípus">
87 |   <xs:sequence>
88 |     <xs:element ref="kapacitás" />
89 |     <xs:element name="cim">
90 |       <xs:complexType>
91 |         <xs:sequence>
92 |           <xs:element name="település" type="xs:string" />
93 |           <xs:element name="utca" type="xs:string" />
94 |           <xs:element name="házszám" type="xs:integer" />
95 |         </xs:sequence>
96 |       </xs:complexType>
97 |     </xs:element>
98 |     <xs:element ref="minősítés" minOccurs="1" maxOccurs="unbounded" />
99 |   </xs:sequence>
100 |   <xs:attribute name="TELEP_ID" type="xs:integer" use="required" />
101 </xs:complexType>
102 <xs:complexType name="tulajdonosTípus">
103 |   <xs:sequence>
104 |     <xs:element ref="név" />
105 |     <xs:element ref="telszám" minOccurs="1" maxOccurs="unbounded" />
106 |     <xs:element ref="adoszam" />
107 |     <xs:element ref="bevétel" />
108 |   </xs:sequence>
109 |   <xs:attribute name="TUL_ID" type="xs:integer" use="required" />
110 |   <xs:attribute name="TUL_TELEP" type="xs:integer" use="required" />
111 </xs:complexType>

```

```

112 <xs:complexType name="autokTípus">
113     <xs:sequence>
114         <xs:element ref="rendszer" />
115         <xs:element ref="modell" minOccurs="1" maxOccurs="unbounded" />
116         <xs:element ref="alvázszer" />
117         <xs:element ref="gyártásiév" />
118     </xs:sequence>
119     <xs:attribute name="AUTO_ID" type="xs:integer" use="required" />
120     <xs:attribute name="AUTO_TELEP" type="xs:integer" use="required" />
121 </xs:complexType>
122 <xs:complexType name="bérlőTípus">
123     <xs:sequence>
124         <xs:element ref="név" />
125         <xs:element ref="telszer" minOccurs="1" maxOccurs="unbounded" />
126         <xs:element ref="email" />
127         <xs:element ref="jogosítványszám" />
128     </xs:sequence>
129     <xs:attribute name="BÉRLŐ_ID" type="xs:integer" use="required" />
130     <xs:attribute name="BÉRLŐ_AUTO" type="xs:integer" use="required" />
131 </xs:complexType>
132 <xs:complexType name="alkalmazottTípus">
133     <xs:sequence>
134         <xs:element ref="név" />
135         <xs:element ref="beosztás" minOccurs="1" maxOccurs="unbounded" />
136         <xs:element ref="életkor" />
137         <xs:element ref="születésiév" />
138     </xs:sequence>
139     <xs:attribute name="ALK_ID" type="xs:integer" use="required" />
140     <xs:attribute name="ALK_TELEP" type="xs:integer" use="required" />
141 </xs:complexType>
142 <xs:complexType name="képesítésTípus">
143     <xs:sequence>
144         <xs:element ref="bizosítványszám" />
145         <xs:element ref="év" />
146         <xs:element ref="nyelv" />
147         <xs:element ref="típus" />
148     </xs:sequence>
149     <xs:attribute name="KÉPESÍTÉS_ID" type="xs:integer" use="required" />
150 </xs:complexType>
151 <xs:complexType name="bónuszTípus">
152     <xs:sequence>
153         <xs:element ref="extra" />
154     </xs:sequence>
155     <xs:attribute name="BONUSZ_ID" type="xs:integer" use="required" />
156     <xs:attribute name="KÉPESÍTÉS_FK" type="xs:integer" use="required" />
157     <xs:attribute name="ALKALMAZOTT_FK" type="xs:integer" use="required" />
158 </xs:complexType>
159
160 <!-- Gyökérelemről az elemek felhasználása -->
161 <xs:element name="H2Z4X3_autokolcsonzo">
162     <xs:complexType>
163         <xs:sequence>
164             <xs:element name="telep" type="telepTípus" minOccurs="0" maxOccurs="10"/>
165             <xs:element name="tulajdonos" type="tulajdonosTípus" minOccurs="0" maxOccurs="10"/>
166             <xs:element name="autok" type="autokTípus" minOccurs="0" maxOccurs="unbounded"/>
167             <xs:element name="bérlő" type="bérlőTípus" minOccurs="0" maxOccurs="unbounded"/>
168             <xs:element name="alkalmazott" type="alkalmazottTípus" minOccurs="0" maxOccurs="unbounded"/>
169             <xs:element name="bónusz" type="bónuszTípus" minOccurs="0" maxOccurs="unbounded"/>
170             <xs:element name="képesítés" type="képesítésTípus" minOccurs="0" maxOccurs="50"/>
171         </xs:sequence>
172     </xs:complexType>

```

```

174 <!-- Elsődleges kulcsok-->
175 <xs:key name="telep_kulcs">
176 |   <xs:selector xpath="telep"/>
177 |   <xs:field xpath="@TELEP_ID"/>
178 </xs:key>
179 <xs:key name="tulajdonos_kulcs">
180 |   <xs:selector xpath="tulajdonos"/>
181 |   <xs:field xpath="@TUL_ID"/>
182 </xs:key>
183 <xs:key name="autok_kulcs">
184 |   <xs:selector xpath="fellépő"/>
185 |   <xs:field xpath="@AUTO_ID"/>
186 </xs:key>
187 <xs:key name="bérelő_kulcs">
188 |   <xs:selector xpath="bérelő"/>
189 |   <xs:field xpath="@BÉRLŐ_ID"/>
190 </xs:key>
191 <xs:key name="alkalmazott_kulcs">
192 |   <xs:selector xpath="alkalmazott"/>
193 |   <xs:field xpath="@ALK_ID"/>
194 </xs:key>
195 <xs:key name="bonusz_kulcs">
196 |   <xs:selector xpath="bonusz"/>
197 |   <xs:field xpath="@BONUSZ_ID"/>
198 </xs:key>
199 <xs:key name="képsítés_kulcs">
200 |   <xs:selector xpath="képesítés"/>
201 |   <xs:field xpath="@KÉPESÍTÉS_ID"/>
202 </xs:key>

205 <!-- Idegen kulcsok-->
206
207 <xs:keyref name="telep_tulajdonos_kulcs" refer="telep_kulcs">
208 |   <xs:selector xpath="tulajdonos"/>
209 |   <xs:field xpath="@telep"/>
210 </xs:keyref>
211 <xs:keyref name="telep_autok_kulcs" refer="telep_kulcs">
212 |   <xs:selector xpath="autok"/>
213 |   <xs:field xpath="@telep"/>
214 </xs:keyref>
215
216 <xs:keyref name="telep_alkalmazott_kulcs" refer="telep_kulcs">
217 |   <xs:selector xpath="alkalmazott"/>
218 |   <xs:field xpath="@telep"/>
219 </xs:keyref>
220 <xs:keyref name="autok_bérelő_kulcs" refer="autok_kulcs">
221 |   <xs:selector xpath="bérelő"/>
222 |   <xs:field xpath="@autok"/>
223 </xs:keyref>
224 <xs:keyref name="alkalmazott_bonusz_kulcs" refer="alkalmazott_kulcs">
225 |   <xs:selector xpath="bonusz"/>
226 |   <xs:field xpath="@alkalmazott"/>
227 </xs:keyref>
228 <xs:keyref name="képesítés_bonusz_kulcs" refer="képsítés_kulcs">
229 |   <xs:selector xpath="bonusz"/>
230 |   <xs:field xpath="@képesítés"/>
231 </xs:keyref>
232
233
234 <!-- Az 1:1 kapcsolat megvalósítás -->
235 <xs:unique name="telep_tulajdonos_egyegy">
236 |   <xs:selector xpath="tulajdonos"/>
237 |   <xs:field xpath="@ftelep"/>
238 </xs:unique>
239 <xs:unique name="autok_bérelő_egyegy">
240 |   <xs:selector xpath="bérelő"/>
241 |   <xs:field xpath="@autok"/>
242 </xs:unique>
243
244
245 </xs:element>
246
247
248 </xs:schema>

```

2. feladat

A feladat egy DOM program készítése az XML dokumentum - *XMLNeptunkod.xml* – adatainak adminisztrálása alapján

2a) adatolvasás

Az alábbi Java-kód képes az XML-adatok olvasására és feldolgozására (DOM) használatával.

```
package hu.domparse.h2z4x3;

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

import java.io.File;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import javax.xml.parsers.*;
import javax.xml.transform.OutputKeys;
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.*;
import org.xml.sax.SAXException;

public class DomReadH2Z4X3 {

    public static void main(String[] args) {

        //xml read meghívása
        readXMLDocument("XMLh2z4X3.xml");
    }
}
```

```

}

/*
 * Ez a metodus beolvassa
 */
private static void readXMLDocument(String filePath) {
    try {

        File newXMLFile = new File(filePath);
        StreamResult newXmlStream = new StreamResult(newXMLFile);

        Document document = parseXML("./XMLh2z4X3.xml");

        writeDocument(document, newXmlStream);

        System.out.println(formatXML(document)); //strukturáltan kiiratás

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

/*
 * Az alábbi metodus beolvassa az XML dokumentumot
 */
public static Document parseXML(String fileName) throws ParserConfigurationException,
SAXException, IOException{

    DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();

    DocumentBuilder builder = factory.newDocumentBuilder();

    // XML dokumentum beolvasása és DOM dokumentummá alakítása
    Document document = builder.parse(new File(fileName));

    // metodus meghivas üres szövegek eltávolítására
    cleanDocument(document.getDocumentElement());

    return document;
}

```

```

/*
 *Delete empty lines
 *
 */
private static void cleanDocument(Node root) {
    NodeList nodeList = root.getChildNodes();

    // Delete empty text spaces
    List<Node> toDeleteEmptyTextList = new ArrayList<>();

    //veigjarni a gyoker elemeket
    for (int i = 0; i < nodeList.getLength(); i++) {

        //megnezi hogy a textnode ures-e
        if (nodeList.item(i).getNodeType() == Node.TEXT_NODE &&
nodeList.item(i).getTextContent().strip().isEmpty()) {

            toDeleteEmptyTextList.add(nodeList.item(i)); //hozzadja

        } else {
            cleanDocument(nodeList.item(i));
        }
    }

    for (Node node : toDeleteEmptyTextList) {

        root.removeChild(node);
    }
}

/*
 * Az alábbi metodus kiirja file-ba az xml dokumentumot
 */
public static void writeDocument(Document document, StreamResult output){
    try {

        TransformerFactory transformerFactory = TransformerFactory.newInstance();

        Transformer transformer = transformerFactory.newTransformer();

        transformer.setOutputProperty(OutputKeys.ENCODING, "UTF-8");

        transformer.setOutputProperty(OutputKeys.INDENT, "yes");

        // DOMletrehozás dokumentumból
        DOMSource source = new DOMSource(document);

```



```

        transformer.transform(source, output);

    } catch (Exception e) {

        e.printStackTrace();

    }
}

/*
 * XML formazo metodus
 */
public static String formatXML(Document document) {

    return "<?xml version=\"" + document.getXmlVersion() + "\" encoding=\"" +
document.getXmlEncoding() + "\" ?>\n" +
        formatElement(document.getDocumentElement(), 0);

}

/*
 * Az xml kiirast formazza
 */
public static String formatElement(Node node, int indent) {

    if (node.getNodeType() != Node.ELEMENT_NODE) {
        return "";
    }

    StringBuilder output = new StringBuilder();

    output.append(getIndent(indent)).append("<").append(((Element) node).getTagName());

    if (node.hasAttributes()) {
        for (int i = 0; i < node.getAttributes().getLength(); i++) {

            Node attribute = node.getAttributes().item(i);
            output.append("
").append(attribute.getNodeName()).append("=\"").append(attribute.getNodeValue()).append("\"");

        }
    }

    //gyerekek lekerdezese
    NodeList children = node.getChildNodes();

```

```

        if (children.getLength() == 1 && children.item(0).getNodeTypes() == Node.TEXT_NODE) {

output.append(">").append(children.item(0).getTextContent().trim()).append("</").append(((Element)
node).getTagName()).append(">\n");
        } else {

            output.append(">\n"); //zarotag

            for (int i = 0; i < children.getLength(); i++) {

                output.append(formatElement(children.item(i), indent + 1));
            }

            // zarotag hozzadassa
            output.append(getIndent(indent)).append("</").append(((Element)
node).getTagName()).append(">\n");
        }

        return output.toString();
    }

    private static String getIndent(int indent) {

        StringBuilder indentation = new StringBuilder();

        for (int i = 0; i < indent; i++) {
            indentation.append(" ");
        }
        return indentation.toString();
    }
}

```



```

        if ("nyelv".equals(childNode.getNodeName())) {

            if ("magyar".equals(childNode.getTextContent())) {
                childNode.setTextContent("német");
            }
        }
    }
}

//2. Bevétele növelése 130 000ft-al Cset Elek-nek.
NodeList telepBevetellList = document.getElementsByTagName("tulajdonos");
for (int temp = 0; temp < telepBevetellList.getLength(); temp++) {
    Node node = telepBevetellList.item(temp);

    if (node.getNodeType() == Node.ELEMENT_NODE) {

        NodeList childNodes = node.getChildNodes();

        for (int j = 0; j < childNodes.getLength(); j++) {
            Node childNode = childNodes.item(j);

            if ("név".equals(childNode.getNodeName())) {
                if ("Cset Elek".equals(childNode.getTextContent())) {

                    for (int k = 0; k < childNodes.getLength(); k++) {
                        childNode = childNodes.item(k);
                        if ("bevétele".equals(childNode.getNodeName())) {
                            childNode.setTextContent("1930000");
                        }
                    }
                }
            }
        }
    }
}

```

```

//3. Ahol a telep kapacitás 5 módosítsuk 6ra
NodeList telepKapacatisList = document.getElementsByTagName("telep");
for (int temp = 0; temp < telepKapacatisList.getLength(); temp++) {
    Node node = telepKapacatisList.item(temp);

    if (node.getNodeType() == Node.ELEMENT_NODE) {

        NodeList childNodes = node.getChildNodes();

        for (int j = 0; j < childNodes.getLength(); j++) {
            Node childNode = childNodes.item(j);

```

```

        if ("kapacitás".equals(childNode.getNodeName())) {

            if ("5".equals(childNode.getTextContent())) {
                childNode.setTextContent("6");
            }

        }
    }
}

```

//4. Az 5. alkalmazott munkahelyének (telepid) módosítása

```

NodeList alkalmazottNList = document.getElementsByTagName("alkalmazott");
for (int temp = 0; temp < alkalmazottNList.getLength(); temp++) {
    Node node = alkalmazottNList.item(temp);

    NamedNodeMap attribute = node.getAttributes();
    Node nodeAttributeID = attribute.getNamedItem("ALK_ID");
    Node nodeAttributeFK = attribute.getNamedItem("ALK_TELEP");

    if (nodeAttributeID.getTextContent().equals("5")) {
        nodeAttributeFK.setTextContent("4");
    }
}

```

//5. Ahol a bónusz kevesebb mint 20000 , az értéket módosítsuk 15000-re

```

NodeList bonuszList = document.getElementsByTagName("bonusz");
for (int temp = 0; temp < bonuszList.getLength(); temp++) {
    Node node = bonuszList.item(temp);

    if (node.getNodeType() == Node.ELEMENT_NODE) {

        NodeList childNodes = node.getChildNodes();

        for (int j = 0; j < childNodes.getLength(); j++) {

            Node childNode = childNodes.item(j);

            if ("extra".equals(childNode.getNodeName())) {

                if (Integer.valueOf(childNode.getTextContent()) <= 20000) {

                    childNode.setTextContent("15000");
                }
            }
        }
    }
}

```

```
//a modositott adatok kiirasa konzolra
```

```
TransformerFactory transformerFactory = TransformerFactory.newInstance();
```

```
Transformer transformer = transformerFactory.newTransformer();
```

```
DOMSource source = new DOMSource(document);
```

```
System.out.println("---- Modositott fajl -----");
```

```
StreamResult consoleResult = new StreamResult(System.out);
```

```
transformer.transform(source, consoleResult);
```

```
} catch (Exception e ) {
```

```
    e.printStackTrace();
```

```
}
```

```
}
```

```
}
```

2c) adatlekérdezés

Az alábbi java kód xml fájlból beolvasott adatok meghatározott lekérdezésére szolgál.

A lekérdezések gyakran egyéb feltételekhez kötöttek, ezek az attribútumok vagy tulajdonságok értékeihez vannak kötve.

A lekérdezések eredményeit futás után kiíratjuk a konzolra.

A következő 5 lekérdezés van megvalósítva a programban:

1. Az A minősítésű telepek kapacitásának lekérdezése
2. Az 1500000ft-nál nagyobb bevételű telep tulajdonosának kiírása
3. A '2'-es ID-jű telep autóinak kiírása amelyek 2020 vagy attól kisebb évjáratúak.
4. Írjuk ki a 'Git Áron' elérhetőségeit
5. A 2-es telep alkalmazottainak az adatai

```
package hu.domparse.h2z4x3;

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

public class DomQueryH2Z4X3 {

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

            //xml file megnyitasa adatok beolvasasahoz
            File file = new File("XMLh2z4x3.xml");

            //dokumentum létrehozasa a beolvasott file-ból
            DocumentBuilder documentBuilder = DocumentBuilderFactory.newInstance().newDocumentBuilder();
            Document document = documentBuilder.parse(file);

            //dokumentum normalizálása
            document.getDocumentElement().normalize();

            lekerdezes1(document);
            lekerdezes2(document);
            lekerdezes3(document);
            lekerdezes5(document);
            lekerdezes6(document);
```

```

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

}

public static void lekerdezes1(Document document) {

    try {

        System.out.println("\n1. Az A minősítésű telepek kapacitásának lekérdezése");
        NodeList nList1 = document.getElementsByTagName("telep");

        for (int i = 0; i < nList1.getLength(); i++) {

            Node nNode = nList1.item(i);

            if (nNode.getNodeType() == Node.ELEMENT_NODE) {

                Element elem = (Element) nNode;
                NodeList childNodes = nNode.getChildNodes();

                for (int j = 0; j < childNodes.getLength(); j++) {

                    Node childNode = childNodes.item(j);

                    if ("minősítés".equals(childNode.getNodeName())) {

                        if ("A".equals(childNode.getTextContent())) {

                            System.out.println("");

                            String telepid = elem.getAttribute("TELEP_ID");
                            Node node1 = elem.getElementsByTagName("kapacitás").item(0);
                            String node1Text = node1.getTextContent();

                            System.out.println("<telep TELEP_ID = " + telepid + " >");
                            System.out.println(" <kapacitás> " + node1Text + " </kapacitás>");
                            System.out.println("</telep>");

                        }

                    }

                }

            }

        }

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

    } catch (NullPointerException e) {

```



```

        e.printStackTrace();
    }

}

public static void lekerdezes2(Document document) {

    try {

        System.out.println("\n2. Az 1500000ft-nál nagyobb bevételű telep tulajdonosának kiírása");
        NodeList nList2 = document.getElementsByTagName("tulajdonos");
        for (int i = 0; i < nList2.getLength(); i++) {

            Node nNode = nList2.item(i);

            if (nNode.getNodeType() == Node.ELEMENT_NODE) {

                Element elem = (Element) nNode;
                NodeList childNodes = nNode.getChildNodes();

                NamedNodeMap attr = nNode.getAttributes();

                if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                    for (int j = 0; j < childNodes.getLength(); j++) {

                        Node childNode = childNodes.item(j);

                        if ("bevétel".equals(childNode.getNodeName())) {

                            if (Integer.valueOf(childNode.getTextContent()) > 1500000 ) {
                                System.out.println("");

                                String tulajdonosid = elem.getAttribute("TUL_ID");

                                Node node1 = elem.getElementsByTagName("név").item(0);
                                String node1Text = node1.getTextContent();

                                System.out.println("<tulajdonos Tulajdonos_ID = > " + tulajdonosid
+ ">");

                                System.out.println("  <név> " + node1Text + "</név>");
                                System.out.println("</tulajdonos>");

                            }

                        }

                    }

                }

            }

        }

        System.out.println("*****");
    }
}

```

```

    } catch (NullPointerException e) {
        e.printStackTrace();
    }
}

public static void lekerdezes3(Document document) {

    try {

        System.out.println("\n3.: A '2'-es ID-jú telep autóinak kiíratása amelyek 2020 vagy attól
kisebb évjáratúak.");
        NodeList nList3 = document.getElementsByTagName("autok");

        for (int i = 0; i < nList3.getLength(); i++) {

            Node nNode = nList3.item(i);

            if (nNode.getNodeType() == Node.ELEMENT_NODE) {

                Element elem = (Element) nNode;
                NodeList childNodes = nNode.getChildNodes();

                NamedNodeMap attr = nNode.getAttributes();
                Node nodeAttribute = attr.getNamedItem("AUTO_TELEP");

                if (nodeAttribute.getTextContent().equals("2")) {
                    for (int j = 0; j < childNodes.getLength(); j++) {

                        Node childNode = childNodes.item(j);

                        if ("gyártásiév".equals(childNode.getNodeName())) {

                            if (Integer.valueOf(childNode.getTextContent()) <= 2020 ) {
                                System.out.println("");

                                String autoid = elem.getAttribute("AUTO_ID");

                                Node node1 = elem.getElementsByTagName("rendszer").item(0);
                                String node1Text = node1.getTextContent();

                                Node node2 = elem.getElementsByTagName("modell").item(0);
                                String node2Text = node2.getTextContent();

                                System.out.println("<autok AUTO_ID = " + autoid + ">");
                                System.out.println("  <rendszer> " + node1Text + " <rendszer>");
                                System.out.println("  <modell> " + node2Text + " </modell> ");
                                System.out.println("</autok>");
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

        }
    }
}

}
System.out.println("*****");

} catch (NullPointerException e) {
    e.printStackTrace();
}

}

public static void lekerdezes5(Document document) {

    try {
        System.out.println("\n5. Írjuk ki a 'Git Áron' elérhetőségeit");
        NodeList nList5 = document.getElementsByTagName("bérló");

        for (int i = 0; i < nList5.getLength(); i++) {

            Node nNode = nList5.item(i);

            if (nNode.getNodeType() == Node.ELEMENT_NODE) {

                Element elem = (Element) nNode;
                NodeList childNodes = nNode.getChildNodes();

                for (int j = 0; j < childNodes.getLength(); j++) {

                    Node childNode = childNodes.item(j);

                    if ("név".equals(childNode.getNodeName())) {

                        if ("Git Áron".equals(childNode.getTextContent())) {
                            System.out.println("");

                            String telepid = elem.getAttribute("BÉRLŐ_ID");

                            Node node1 = elem.getElementsByTagName("telszám").item(0);
                            String node1Text = node1.getTextContent();
                            Node node2 = elem.getElementsByTagName("email").item(0);
                            String node2Text = node2.getTextContent();

                            System.out.println("<bérló BÉRLŐ_ID = " + telepid + " >");
                            System.out.println("  <telszám> " + node1Text + " </telszám>");
                            System.out.println("  <email> " + node2Text + "</email>");
                            System.out.println("</bérló>");
                        }
                    }
                }
            }
        }
    }
}

```

```

        }

    }

}

}

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

} catch (NullPointerException e) {
    e.printStackTrace();
}

}

public static void lekerdezes6(Document document) {

    try {
        System.out.println("\n6. A 2-es telep alkalmazottinak az adatai");
        NodeList nList6 = document.getElementsByTagName("alkalmazott");

        for (int i = 0; i < nList6.getLength(); i++) {

            Node nNode = nList6.item(i);

            if (nNode.getNodeType() == Node.ELEMENT_NODE) {

                Element elem = (Element) nNode;
                NamedNodeMap attr = nNode.getAttributes();
                Node nodeAttrM = attr.getNamedItem("ALK_TELEP");

                if (nodeAttrM.getTextContent().equals("2")) {

                    System.out.println("");

                    String alkalmazottid = elem.getAttribute("ALK_ID");

                    Node node1 = elem.getElementsByTagName("név").item(0);
                    String node1Text = node1.getTextContent();
                    Node node2 = elem.getElementsByTagName("beosztás").item(0);
                    String node2Text = node2.getTextContent();
                    Node node3 = elem.getElementsByTagName("életkor").item(0);
                    String node3Text = node3.getTextContent();
                    Node node4 = elem.getElementsByTagName("születésiév").item(0);
                    String node4Text = node4.getTextContent();

                    System.out.println("<alkalmazott ALKALMAZOTT_ID = " + alkalmazottid +
">");

                    System.out.println("  <név> " + node1Text + " </név>");
                    System.out.println("  <beosztás> " + node2Text + " </beosztás>");
                    System.out.println("  <életkor> " + node3Text + " </életkor>");
                    System.out.println("  <születésiév> " + node4Text + " </születésiév>");
                    System.out.println("</alkalmazott>");
                }
            }
        }
    }
}

```

```
        }

    }

}

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

} catch (NullPointerException e) {
    e.printStackTrace();
}

}
```

2d) adatírás

Az alábbi java kód létrehozza az új XML file tartalmát, majd egy fastruktúrában reprezentálja és kiírja egy új XML fájlba.

```
package hu.domparse.h2z4x3;

import java.io.File;
import java.io.FileOutputStream;
import java.io.OutputStream;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.transform.OutputKeys;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerConfigurationException;
import javax.xml.transform.TransformerException;
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.Node;
import org.w3c.dom.NodeList;

import org.w3c.dom.Document;
import org.w3c.dom.Element;

public class DomWriteH2Z4X3 {
    public static void main(String args[]) {
        try {
            DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
            DocumentBuilder dbBuilder = dbFactory.newDocumentBuilder();
            Document document = dbBuilder.newDocument();

            // root elements megadása
            Element root = document.createElement("autokolcsonzo_H2Z4X3");
            root.setAttribute("xmlns:xs", "http://www.w3.org/2001/XMLSchema-instance");
            root.setAttribute("xs:noNamespaceSchemaLocation", "XMLSchemaH2Z4X3.xsd");
            document.appendChild(root);

            // Telep
            root.appendChild(document.createComment("Telep"));
            addTelep(document, root, "1", "5", "Budapest", "Benczúr_utca", "23", "B");
            addTelep(document, root, "2", "8", "Budapest", "Jókai_utca", "14", "A");
            addTelep(document, root, "3", "6", "Szeged", "Szabadság_utca", "23", "C");
            addTelep(document, root, "4", "6", "Siófok", "Balaton_utca", "45", "A");

            // Tulajdonos
            root.appendChild(document.createComment("Tulajdonos"));
            addTulajdonos(document, root, "1", "1", "Kis János", "067022344555", "12345678-1-11", "1500000" );
            addTulajdonos(document, root, "2", "2", "Cicz Imre", "06205678899", "18384763-2-24", "1234560" );
        }
    }
}
```

```
addTulajdonos(document, root, "3", "3", "Cset Elek", "06703452233", "2345679-3-13", "1800000" );
addTulajdonos(document, root, "4", "4", "Kis Anikó", "06304567788", "12341236-2-31", "1850000" );
```

```
// Autok
```

```
root.appendChild(document.createComment("Autok"));
addAuto(document, root, "1", "1", "PSY-213", "Jaguar_F-TYPE", "JZMMA18P200411817", "2020" );
addAuto(document, root, "2", "2", "RYZ-022", "Mercedes_S63_AMG", "JZMMA18P200411817", "2021" );
addAuto(document, root, "3", "3", "SUV-123", "Volvo_XC-90", "JZMMA18P201111111", "2018");
addAuto(document, root, "4", "4", "SRY-345", "Audi_A5", "JZMMA18P200411817", "2020" );
addAuto(document, root, "5", "4", "PSY-234", "BMW_M5_Competition", "PZMMA18P200434567", "2018" );
```

```
// Berlo
```

```
root.appendChild(document.createComment("Berlo"));
addBerlo(document, root, "1", "2", "Diz Elek", "06703439988", "dizelek@gmail.com", "WW001002" );
addBerlo(document, root, "2", "3", "Farkas Piroska", "06703439988",
"piroksa@gfreemail.hu", "WW001002" );
addBerlo(document, root, "3", "4", "Feles Elek", "06703439988", "felelek@gmail.com", "MM020105" );
addBerlo(document, root, "4", "5", "Git Áron", "06204563322", "gitaron@citromail.hu", "WW001002" );
```

```
// Alkalmazott
```

```
root.appendChild(document.createComment("Alkalmazott"));
addAlkalmazott(document, root, "1", "2", "Hideg Ottó", "Adminisztrátor", "48", "1975" );
addAlkalmazott(document, root, "2", "3", "Kis Eszter", "Üzletvezető", "45", "1978" );
addAlkalmazott(document, root, "3", "1", "Pat Tamás", "supervisor", "35", "1988" );
addAlkalmazott(document, root, "4", "4", "Tra Pista", "marketing_manager", "39", "1984" );
addAlkalmazott(document, root, "5", "2", "Tra Pista", "hr_vezető", "45", "1978" );
```

```
// Bonusz
```

```
root.appendChild(document.createComment("Bonusz"));
addBonus(document, root, "1", "1", "1", "10000");
addBonus(document, root, "2", "2", "2", "25000");
addBonus(document, root, "3", "3", "3", "30000");
```

```
// Kepesites
```

```
root.appendChild(document.createComment("Képsítés"));
addKepesites(document, root, "1", "7245889", "2021", "angol", "A2" );
addKepesites(document, root, "2", "7245867", "2015", "magyar", "B2" );
addKepesites(document, root, "3", "7224589", "2010", "angol", "C1" );
addKepesites(document, root, "4", "7298174", "2011", "angol", "B1" );
```

```
// Documentum felépítése
```

```
TransformerFactory transformerFactory = TransformerFactory.newInstance();
Transformer transformer = transformerFactory.newTransformer();
transformer.setOutputProperty(OutputKeys.ENCODING, "UTF-8");
transformer.setOutputProperty(OutputKeys.INDENT, "yes");
```

```

transformer.setOutputProperty("{https://xml.apache.org/xslt}indent-amount", "2");

// Kimeneti fájl
DOMSource source = new DOMSource(document);
File outputFile = new File("XmlH2Z4X3_out.xml");
StreamResult file = new StreamResult(outputFile);
transformer.transform(source, file);

//Konzolra kiírás
StreamResult console = new StreamResult(System.out);
transformer.transform(source, console);
}catch(Exception e) {
    e.printStackTrace();
}

}

public static void addTelep(Document document, Element root, String id, String kapacitas, String varos,
    String utca, String hazszam, String minosites) {

    // Létrehoz egy "telep" elemet a megadott paraméterekkel.
    Element carstation = document.createElement("telep");
    carstation.setAttribute("TELEP_ID", id);

    // Létrehoz egy "kapacitás" elemet és hozzáadja az értéket.

    Element capacity = createElementValue(document, "kapacitás", kapacitas);
    // Létrehoz egy "cim" elemet és hozzáadja az "irányítószám", "város", "utca_házzsám" elemeket.

    Element address = document.createElement("cim");
    Element city = createElementValue(document, "irányítószám", varos);
    Element street = createElementValue(document, "város", utca);
    Element number = createElementValue(document, "utca_házzsám", hazszam);

    Element quality = createElementValue(document, "mennyiség", minosites);

    // Hozzáadja az elemeket a megfelelő hierarchiában az "telep" elemhez.
    address.appendChild(city);
    address.appendChild(street);
    address.appendChild(number);

    carstation.appendChild(capacity);
    carstation.appendChild(address);
    carstation.appendChild(quality);

    // Hozzáadja az "telep" elemet a gyökérelemhez.
    root.appendChild(carstation);
}

public static void addTulajdonos(Document document, Element root, String id, String fk, String nev,
    String telszam, String adoszam, String bevetel) {

```



```

Element owners = document.createElement("tulajdonos");
owners.setAttribute("TELEP_ID", id);
owners.setAttribute("TUL_TELEP", fk);

// Elem létrehozása a megadott paraméterekkel.
Element name = createElementValue(document, "név", nev);
Element phone = createElementValue(document, "telszám", telszam);
Element taxid = createElementValue(document, "adoszám", adoszam);
Element income = createElementValue(document, "bevétele", bevetel);

owners.appendChild(name);
owners.appendChild(phone);
owners.appendChild(taxid);
owners.appendChild(income);

root.appendChild(owners);
}

public static void addAuto(Document document, Element root, String id, String fk, String rendszam,
    String modell, String alvazszam, String gyartev) {

    Element cars = document.createElement("autok");
    cars.setAttribute("AUTO_ID", id);
    cars.setAttribute("AUTO_TELEP", fk);

    // Elem létrehozása a megadott paraméterekkel.
    Element licence = createElementValue(document, "rendszám", rendszam);
    Element model = createElementValue(document, "modell", modell);
    Element framenummer = createElementValue(document, "alvázszám", alvazszam);
    Element buildyear = createElementValue(document, "gágyrtásiév", gyartev);

    // Hozzáadja az elemeket a megfelelő hierarchiában
    cars.appendChild(licence);
    cars.appendChild(model);
    cars.appendChild(framenummer);
    cars.appendChild(buildyear);

    // Hozzáadja az "telep" elemet a gyökérelemhez.
    root.appendChild(cars);
}

public static void addBerlo(Document document, Element root, String id, String fk, String p1,
    String p2, String p3, String p4) {

    Element cars = document.createElement("bérló");
    cars.setAttribute("BÉRLŐ_ID", id);
    cars.setAttribute("BÉRLŐ_AUTO", fk);

    // Elem létrehozása a megadott paraméterekkel.
    Element name = createElementValue(document, "név", p1);
    Element phone = createElementValue(document, "telszám", p2);
    Element email = createElementValue(document, "email", p3);
    Element income = createElementValue(document, "jogosítványszám", p4);

```

```

        // Hozzáadja az elemeket a megfelelő hierarchiában
cars.appendChild(name);
cars.appendChild(phone);
cars.appendChild(email);
cars.appendChild(income);

        // Hozzáadja az "telep" elemet a gyökérelemhez.
root.appendChild(cars);
}

public static void addAlkalmazott(Document document, Element root, String id, String fk, String p1,
        String p2, String p3, String p4) {

    Element cars = document.createElement("alkalmazottak");
    cars.setAttribute("AUTO_ID", id);
    cars.setAttribute("AUTO_TELEP", fk);

    // Elem létrehozása a megadott paraméterekkel.
    Element value1 = createElementValue(document, "név", p1);
    Element value2 = createElementValue(document, "beosztás", p2);
    Element value3 = createElementValue(document, "életkor", p3);
    Element value4 = createElementValue(document, "születésiév", p4);

    // Hozzáadja az elemeket a megfelelő hierarchiában
cars.appendChild(value1);
cars.appendChild(value2);
cars.appendChild(value3);
cars.appendChild(value4);

    // Hozzáadja az "telep" elemet a gyökérelemhez.
root.appendChild(cars);
}

public static void addBonus(Document document, Element root, String id, String div_fk,
        String kepesites_fk, String bonus) {
    Element extra = document.createElement("bonusz");
    extra.setAttribute("BONUSZ_ID", id);
    extra.setAttribute("ALKALMAZOTT_FK", div_fk);
    extra.setAttribute("KÉPESÍTÉS_FK", kepesites_fk);
    // Elem létrehozása a megadott paraméterekkel.
    Element _date = createElementValue(document, "mikortól", bonus);
    extra.appendChild(_date);

    // Hozzáadja az "telep" elemet a gyökérelemhez.
root.appendChild(extra);
}

public static void addKepesites(Document document, Element root, String id, String p1,
        String p2, String p3, String p4) {

    Element certificate = document.createElement("képsítés");
    certificate.setAttribute("AUTO_ID", id);

    // Elem létrehozása a megadott paraméterekkel.

```

```
Element value1 = createElementValue(document, "kapacitás", p1);
Element value2 = createElementValue(document, "kapacitás", p2);
Element value3 = createElementValue(document, "kapacitás", p3);
Element value4 = createElementValue(document, "kapacitás", p4);

// Hozzáadja az elemeket a megfelelő hierarchiában
certificate.appendChild(value1);
certificate.appendChild(value2);
certificate.appendChild(value3);
certificate.appendChild(value4);

// Hozzáadja az "telep" elemet a gyökérelemhez.
root.appendChild(certificate);
}

public static Element createElementValue(Document doc, String name, String value) {
    Element elem = doc.createElement(name);
    elem.appendChild(doc.createTextNode(value));
    return elem;
}
}
```