

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	16
2c) adatlekérdezés	25
2d) adatírás	25

A feladat leírása:

A feladat egy autókölcsönző vállalkozáshoz kapcsolódik, amely több telephellyel rendelkezik országszerte. A telephelyeken különböző autók találhatók 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 telepnek 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 szerezhetnek plusz képesítéseket melyekkel növelhetik a fizetésüket. A különböző juttatások 10 000- 20 000- 30 000 Ft összegekben oszlanak meg. A képesítéshez különböző szint és nyelv is tartozik melyek tovább befolyásolják a juttatás mértékét. Egy alkalmazott több képesítéssel is rendelkezhet. A képesítések 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)
- TelSzáma (Tulajdonos telefonszáma)
- TulID (primary key)

Telep:

- Kapacitás (a tárolható autók száma)
- Cím (összetett tulajdonság, tartalmazza a települést, utca, házszámot)
- 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)
- TelSzáma (A bérlő telefon száma)
- Email (A bérlő email címe)
- Jogosítvány szám (Tualjdonos telefonszáma)
- BérlőID (primary key)

Alkalmazottak:

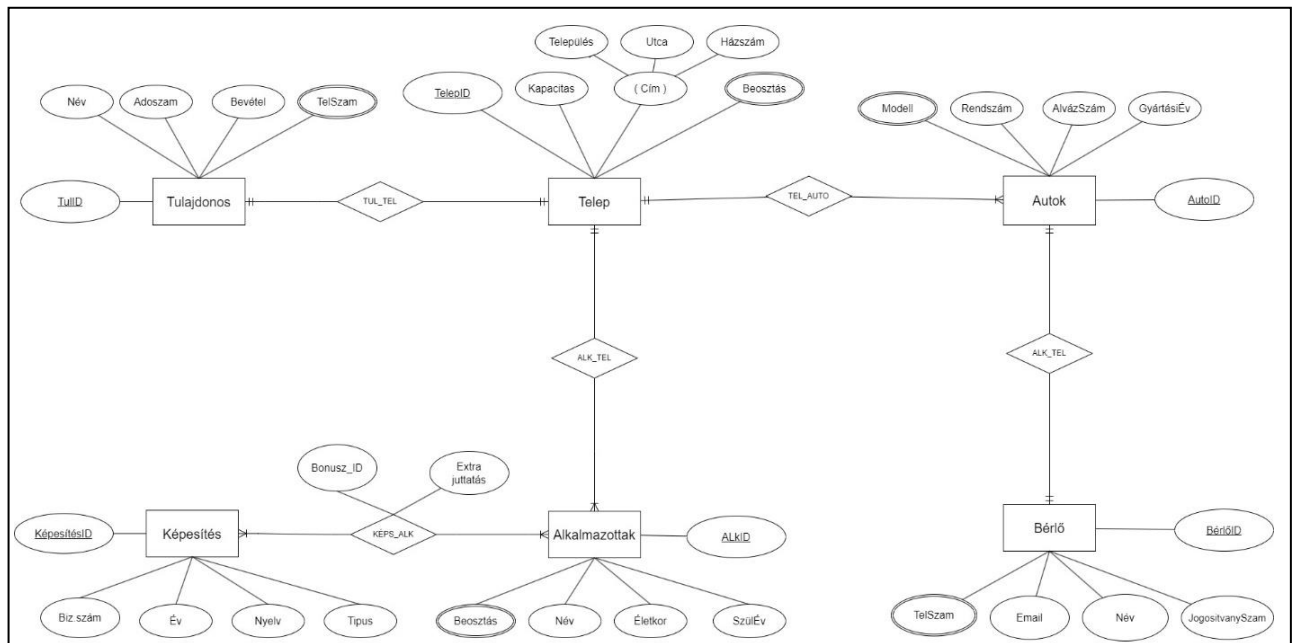
- Név (alkalmazott neve)
- Beosztás (a feladatkör amit az alkalmazott ellát)
- SzülÉv (Az alkalmazott születési éve)
- Életkor (Az alkalmazott életkora)
- AlkID (primary key)

Képesítés:

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

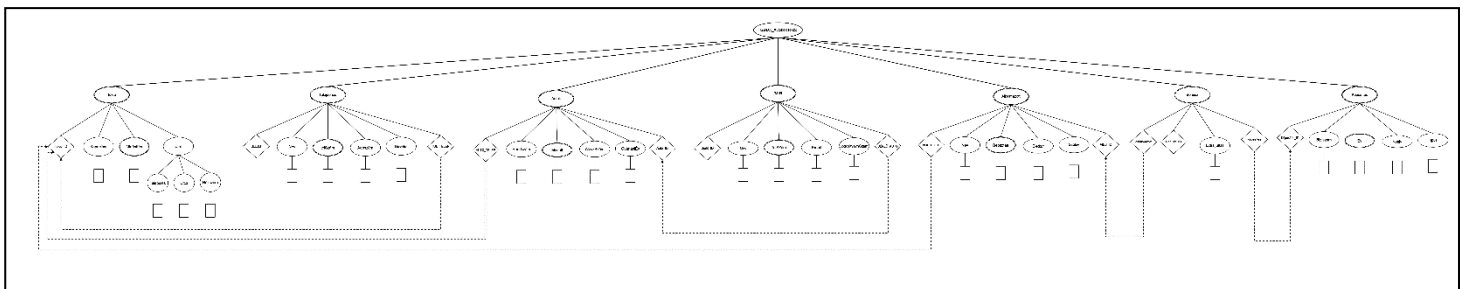
Bonusz:

- 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: (Ide kerül az XML kódja!)

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="bérlő_kulcs">
188 |   <xs:selector xpath="bérlő"/>
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_bérlő_kulcs" refer="autok_kulcs">
221 |   <xs:selector xpath="bérlő"/>
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_bérlő_egyegy">
240 |   <xs:selector xpath="bérlő"/>
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.

```
1 package hu.domparsed.h2z4x3;
2
3 import java.io.File;
4 import java.io.IOException;
5
6 import javax.xml.parsers.DocumentBuilder;
7 import javax.xml.parsers.DocumentBuilderFactory;
8 import javax.xml.parsers.ParserConfigurationException;
9 import javax.xml.transform.Transformer;
10 import javax.xml.transform.TransformerException;
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.Document;
16 import org.w3c.dom.Element;
17 import org.w3c.dom.Node;
18 import org.w3c.dom.NodeList;
19 import org.xml.sax.SAXException;
20
21 public class DomReadH2Z4X3 {
22
23     Run | Debug
24     public static void main(String[] args) throws SAXException, IOException, ParserConfigurationException, TransformerException {
25
26         //xml file megnyitasa adatok beolvasahoz
27         File xmlFile = new File(pathname:"XMLh2z4x3.xml");
28
29         //dokumentum létrehozasa
30         DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
31         DocumentBuilder dBuilder = factory.newDocumentBuilder();
32
33         //dokumentum létrehozasa a beolvasott filebol majd normalizalasa
34         Document document = dBuilder.parse(xmlFile);
35         document.getDocumentElement().normalize();
36
37         System.out.println("Root element: " + document.getDocumentElement().getNodeName());
38
39
40         //beolvasasi metodusok
41         getTelep(document);
42         getTulajdonos(document);
43         getAutok(document);
44         getBerlo(document);
45         getAlkalmazott(document);
46         getBonusz(document);
47         getKepesites(document);
48
49
50         //a megadott adatok kiiras txt fileba
51
52         File modFile = new File(pathname:"XMLH2Z4X3.txt");
53
54         TransformerFactory transformerFactory = TransformerFactory.newInstance();
55         Transformer transformer = transformerFactory.newTransformer();
56
57         DOMSource source = new DOMSource(document);
58
59         StreamResult resultModFile = new StreamResult(modFile);
60         transformer.transform(source, resultModFile );
61
62
63     }
64 }
```

```

65 public static void getTelep(Document doc) {
66
67     NodeList nList = doc.getElementsByTagName(tagname:"telep");
68
69     System.out.println(x:"\nTelep:\n-----");
70
71     for (int i = 0; i < nList.getLength(); i++) {
72
73         Node nNode = nList.item(i);
74
75         System.out.println("\nCurrent Element: " + nNode.getNodeName());
76
77         if (nNode.getNodeType()==Node.ELEMENT_NODE) {
78
79             Element elem = (Element) nNode;
80             String telepid = elem.getAttribute(name:"TELEP_ID");
81
82             Node node1 = elem.getElementsByTagName(name:"kapacitás").item(index:0);
83             String node1Text = node1.getTextContent();
84
85             Node node2 = elem.getElementsByTagName(name:"település").item(index:0);
86             String node2Text = node2.getTextContent();
87
88             Node node3 = elem.getElementsByTagName(name:"utca").item(index:0);
89             String node3Text = node3.getTextContent();
90
91             Node node5 = elem.getElementsByTagName(name:"házszám").item(index:0);
92             String node5Text = node5.getTextContent();
93
94             Node node6 = elem.getElementsByTagName(name:"minősítés").item(index:0);
95             String node6Text = node6.getTextContent();
96
97             System.out.println("Rendelés id: " + telepid);
98
99
100            System.out.println("Kapacitás: " + node1Text);
101
102            System.out.println(" <település>: " + node2Text + "</település>");
103
104            System.out.println(" <utca>: " + node3Text+"</utca>");
105
106            System.out.println(" <házszám>: " + node5Text + "</házszám>");
107
108            System.out.println("<minősítés>: " + node6Text+"</minősítés>");
109        }
110    }
111 }

```



```

117 public static void getTulajdonos(Document doc) {
118
119     NodeList nList = doc.getElementsByTagName(tagname:"tulajdonos");
120
121     System.out.println(x:"\nTulajdonos:\n-----");
122
123     for (int i = 0; i < nList.getLength(); i++) {
124
125         Node nNode = nList.item(i);
126
127         //System.out.println("\nCurrent Element: " + nNode.getNodeName());
128         System.out.println(x:"\n");
129
130         if (nNode.getNodeType()==Node.ELEMENT_NODE) {
131
132             Element elem = (Element) nNode;
133             String tulajdonosid = elem.getAttribute(name:"TUL_ID");
134             String fk = elem.getAttribute(name:"TUL_TELEP");
135
136             Node node1 = elem.getElementsByTagName(name:"név").item(index:0);
137             String node1Text = node1.getTextContent();
138
139             Node node2 = elem.getElementsByTagName(name:"telszám").item(index:0);
140             String node2Text = node2.getTextContent();
141
142             Node node3 = elem.getElementsByTagName(name:"adoszam").item(index:0);
143             String node3Text = node3.getTextContent();
144
145             Node node5 = elem.getElementsByTagName(name:"bevétele").item(index:0);
146             String node5Text = node5.getTextContent();
147
148
149             System.out.println("Tulajdonos_ID: " + tulajdonosid);
150             System.out.println("TUL_TELEP FK: " + fk);
151
152             System.out.println(" <Név>:" + node1Text + "</Név>:");
153
154             System.out.println(" <telefonszám> " + node2Text + " </település>");
155
156             System.out.println(" <adószám> " + node3Text+" </utca>");
157
158             System.out.println(" <bevétele> " + node5Text + " </házzám>");
159
160         }
161     }
162 }
163

```



```

169 public static void getAutok(Document doc) {
170
171     NodeList nList = doc.getElementsByTagName(tagname:"autok");
172
173     System.out.println(x:"\nAutok:\n-----");
174
175     for (int i = 0; i < nList.getLength(); i++) {
176
177         Node nNode = nList.item(i);
178
179         //System.out.println("\nCurrent Element: " + nNode.getNodeName());
180         System.out.println(x:"\n");
181
182         if (nNode.getNodeType()==Node.ELEMENT_NODE) {
183
184             Element elem = (Element) nNode;
185             String id = elem.getAttribute(name:"AUTO_ID");
186             String fk = elem.getAttribute(name:"AUTO_TELEP");
187
188             Node node1 = elem.getElementsByTagName(name:"rendszer").item(index:0);
189             String node1Text = node1.getTextContent();
190
191             Node node2 = elem.getElementsByTagName(name:"modell").item(index:0);
192             String node2Text = node2.getTextContent();
193
194             Node node3 = elem.getElementsByTagName(name:"alvázszám").item(index:0);
195             String node3Text = node3.getTextContent();
196
197             Node node5 = elem.getElementsByTagName(name:"gyártásiév").item(index:0);
198             String node5Text = node5.getTextContent();
199
200
201
202             System.out.println("AUTO_ID: " + id);
203             System.out.println("AUTO_TELEP FK: " + fk);
204
205             System.out.println(" <rendszer>:" + node1Text + "</rendszer>");
206
207             System.out.println(" <modell> " + node2Text + " </modell>");
208
209             System.out.println(" <alvázszám> " + node3Text+" </alvázszám>");
210
211             System.out.println(" <gyártásiév> " + node5Text + " </gyártásiév>");
212
213         }
214     }
215 }
216

```

```
220 public static void getBerlo(Document doc) {
221
222     NodeList nList = doc.getElementsByTagName(tagname:"bérló");
223
224     System.out.println(x:"\nBérlok:\n-----");
225
226     for (int i = 0; i < nList.getLength(); i++) {
227
228         Node nNode = nList.item(i);
229
230         //System.out.println("\nCurrent Element: " + nNode.getNodeName());
231         System.out.println(x:"\n");
232
233         if (nNode.getNodeType()==Node.ELEMENT_NODE) {
234
235             Element elem = (Element) nNode;
236             String id = elem.getAttribute(name:"BÉRLŐ_ID");
237             String fk = elem.getAttribute(name:"BÉRLŐ_AUTO");
238
239             Node node1 = elem.getElementsByTagName(name:"név").item(index:0);
240             String node1Text = node1.getTextContent();
241
242             Node node2 = elem.getElementsByTagName(name:"telszám").item(index:0);
243             String node2Text = node2.getTextContent();
244
245             Node node3 = elem.getElementsByTagName(name:"email").item(index:0);
246             String node3Text = node3.getTextContent();
247
248             Node node5 = elem.getElementsByTagName(name:"jogosítványszám").item(index:0);
249             String node5Text = node5.getTextContent();
250
251
252
253             System.out.println("AUTO_ID: " + id);
254             System.out.println("AUTO_TELEP FK: " + fk);
255
256             System.out.println(" <név>:" + node1Text + "</név>");
257
258             System.out.println(" <telszám> " + node2Text + " </telszám>");
259
260             System.out.println(" <email> " + node3Text+" </email>");
261
262             System.out.println(" <jogosítványszám> " + node5Text + " </jogosítványszám>");
263
264         }
265
266     }
```

```

270 public static void getAlkalmazott(Document doc) {
271
272     NodeList nList = doc.getElementsByTagName(tagname:"alkalmazott");
273
274     System.out.println(x:"\nAlkalmazottak:\n-----");
275
276     for (int i = 0; i < nList.getLength(); i++) {
277
278         Node nNode = nList.item(i);
279
280         //System.out.println("\nCurrent Element: " + nNode.getNodeName());
281         System.out.println(x:"\n");
282
283         if (nNode.getNodeType()==Node.ELEMENT_NODE) {
284
285             Element elem = (Element) nNode;
286             String id = elem.getAttribute(name:"ALK_ID");
287             String fk = elem.getAttribute(name:"ALK_ID");
288
289             Node node1 = elem.getElementsByTagName(name:"név").item(index:0);
290             String node1Text = node1.getTextContent();
291
292             Node node2 = elem.getElementsByTagName(name:"beosztás").item(index:0);
293             String node2Text = node2.getTextContent();
294
295             Node node3 = elem.getElementsByTagName(name:"életkor").item(index:0);
296             String node3Text = node3.getTextContent();
297
298             Node node5 = elem.getElementsByTagName(name:"születésiév").item(index:0);
299             String node5Text = node5.getTextContent();
300
301
302
303             System.out.println("ALK_ID: " + id);
304             System.out.println("ALK_ID FK: " + fk);
305
306             System.out.println(" <név>:" + node1Text + "</név>");
307
308             System.out.println(" <beosztás> " + node2Text + " </beosztás>");
309
310             System.out.println(" <életkor> " + node3Text+" </életkor>");
311
312             System.out.println(" <születésiév> " + node5Text + " </születésiév>");
313
314
315         }

```

```

320 }
321 public static void getBonusz(Document doc) {
322
323     NodeList nList = doc.getElementsByTagName(tagname:"bonusz");
324
325     System.out.println(x:"\nBonusz:\n-----");
326
327     for (int i = 0; i < nList.getLength(); i++) {
328
329         Node nNode = nList.item(i);
330
331         //System.out.println("\nCurrent Element: " + nNode.getNodeName());
332         System.out.println(x:"\n");
333
334         if (nNode.getNodeType()==Node.ELEMENT_NODE) {
335
336             Element elem = (Element) nNode;
337             String id = elem.getAttribute(name:"BONUSZ_ID");
338             String fk = elem.getAttribute(name:"ALKALMAZOTT_FK");
339             String fk2 = elem.getAttribute(name:"KÉPESÍTÉS_FK");
340
341             Node node1 = elem.getElementsByTagName(name:"extra").item(index:0);
342             String node1Text = node1.getTextContent();
343
344
345
346
347             System.out.println("ALK_ID: " + id);
348             System.out.println("ALK_ID FK: " + fk);
349             System.out.println("ALK_ID FK: " + fk2);
350
351             System.out.println(" <extra>" + node1Text + "</extra>");
352
353         }
354     }
355 }
356

```

```

360 public static void getKepesites(Document doc) {
361
362     NodeList nList = doc.getElementsByTagName(tagname:"képesítés");
363
364     System.out.println(x:"\nKépesítés:\n-----");
365
366     for (int i = 0; i < nList.getLength(); i++) {
367
368         Node nNode = nList.item(i);
369
370         //System.out.println("\nCurrent Element: " + nNode.getNodeName());
371         System.out.println(x:"\n");
372
373         if (nNode.getNodeType()==Node.ELEMENT_NODE) {
374
375             Element elem = (Element) nNode;
376             String id = elem.getAttribute(name:"KÉPESÍTÉS_ID");
377
378
379             Node node1 = elem.getElementsByTagName(name:"bizosítványszám").item(index:0);
380             String node1Text = node1.getTextContent();
381
382             Node node2 = elem.getElementsByTagName(name:"év").item(index:0);
383             String node2Text = node2.getTextContent();
384
385             Node node3 = elem.getElementsByTagName(name:"nyelv").item(index:0);
386             String node3Text = node3.getTextContent();
387
388             Node node5 = elem.getElementsByTagName(name:"típus").item(index:0);
389             String node5Text = node5.getTextContent();
390
391
392             System.out.println("KÉPESÍTÉS_ID: " + id);
393
394             System.out.println(" <bizosítványszám>:" + node1Text + "</bizosítványszám>");
395
396             System.out.println(" <év> " + node2Text + " </év>");
397
398             System.out.println(" <nyelv> " + node3Text+" </nyelv>");
399
400             System.out.println(" <típus> " + node5Text + " </típus>");
401
402         }
403     }
404 }
405

```

2b) adatmódosítás (kód – comment együtt) – fájlnev: *DOMModifyNeptunkod.java*

Az alábbi Java kód egy DOM alapú XML fájl módosítást valósít meg. A DOM segítségével egy XML fájl tartalmát egy fastruktúrában jeleníti meg, és lehetőséget ad a fájl elemeinek navigálására, módosítására.

```
1  package hu.domparse.h2z4x3;
2
3  import java.io.File;
4  import javax.xml.parsers.DocumentBuilder;
5  import javax.xml.parsers.DocumentBuilderFactory;
6  import javax.xml.transform.Transformer;
7  import javax.xml.transform.TransformerFactory;
8  import javax.xml.transform.dom.DOMSource;
9  import javax.xml.transform.stream.StreamResult;
10
11 import org.w3c.dom.Document;
12 import org.w3c.dom.NamedNodeMap;
13 import org.w3c.dom.Node;
14 import org.w3c.dom.NodeList;
15
16 public class DomModifyH2Z4X3 {
17
18     Run | Debug
19     public static void main(String[] args) {
20
21         try {
22
23             //xml file megnyitása adatok beolvasásához
24             File file = new File(pathname:"XMLh2z4x3.xml");
25
26             //dokumentum létrehozása a beolvasott fileból
27             DocumentBuilder documentBuilder = DocumentBuilderFactory.newInstance().newDocumentBuilder();
28             Document document = documentBuilder.parse(file);
29
30             //dokumentum normalizálás
31             document.getDocumentElement().normalize();
32
33             //1. képesítés nyelv módosítása magyarról németre
34
35             NodeList képesitesNyelvList = document.getElementsByTagName(tagname:"képesítés");
36             for (int temp = 0; temp < képesitesNyelvList.getLength(); temp++) {
37                 Node node = képesitesNyelvList.item(temp);
38
39                 if (node.getNodeType() == Node.ELEMENT_NODE) {
40
41                     NodeList childNodes = node.getChildNodes();
42
43                     for (int j = 0; j < childNodes.getLength(); j++) {
44                         Node childNode = childNodes.item(j);
45
46                         if ("nyelv".equals(childNode.getNodeName())) {
47                             if ("magyar".equals(childNode.getTextContent())) {
48                                 childNode.setTextContent(textContent:"német");
49                             }
50                         }
51                     }
52                 }
53             }
54         }
55     }
56 }
```

```

56
57
58 //2. Bevétele növelése 130 000ft-al Cset Elek-nek.
59 NodeList telepBevetellist = document.getElementsByTagName(tagname:"tulajdonos");
60 for (int temp = 0; temp < telepBevetellist.getLength(); temp++) {
61     Node node = telepBevetellist.item(temp);
62
63     if (node.getNodeType() == Node.ELEMENT_NODE) {
64
65         NodeList childNodes = node.getChildNodes();
66
67         for (int j = 0; j < childNodes.getLength(); j++) {
68             Node childNode = childNodes.item(j);
69
70             if ("név".equals(childNode.getNodeName())) {
71                 if ("Cset Elek".equals(childNode.getTextContent())) {
72
73                     for (int k = 0; k < childNodes.getLength(); k++) {
74                         childNode = childNodes.item(k);
75                         if ("bevétele".equals(childNode.getNodeName())) {
76                             childNode.setTextContent(textContent:"1930000");
77                         }
78                     }
79                 }
80             }
81         }
82     }
83 }
84

```

```

85
86 //3. Ahol a telep kapacitas 5 módosítsuk 6ra
87 NodeList telepKapacatisList = document.getElementsByTagName(tagname:"telep");
88 for (int temp = 0; temp < telepKapacatisList.getLength(); temp++) {
89     Node node = telepKapacatisList.item(temp);
90
91     if (node.getNodeType() == Node.ELEMENT_NODE) {
92
93         NodeList childNodes = node.getChildNodes();
94
95         for (int j = 0; j < childNodes.getLength(); j++) {
96             Node childNode = childNodes.item(j);
97
98             if ("kapacitás".equals(childNode.getNodeName())) {
99
100                 if ("5".equals(childNode.getTextContent())) {
101                     childNode.setTextContent(textContent:"6");
102                 }
103             }
104         }
105     }
106 }
107

```

```

108
109
110 //4. Az 5. alkalmazott munkahelyének (telepid) módosítása
111 NodeList alkalmazottNList = document.getElementsByTagName(tagname:"alkalmazott");
112 for (int temp = 0; temp < alkalmazottNList.getLength(); temp++) {
113     Node node = alkalmazottNList.item(temp);
114
115     NamedNodeMap attribute = node.getAttributes();
116     Node nodeAttributeID = attribute.getNamedItem(name:"ALK_ID");
117     Node nodeAttributeFK = attribute.getNamedItem(name:"ALK_TELEP");
118
119     if (nodeAttributeID.getTextContent().equals(anObject:"5")) {
120         nodeAttributeFK.setTextContent(textContent:"4");
121     }
122 }
123

```

```

124
125 //5. Ahol a bónusz kevesebb mint 20000 , az értéket módosítsuk 15000-re
126 NodeList bonuszList = document.getElementsByTagName(tagname:"bonusz");
127 for (int temp = 0; temp < bonuszList.getLength(); temp++) {
128     Node node = bonuszList.item(temp);
129
130     if (node.getNodeType() == Node.ELEMENT_NODE) {
131
132         NodeList childNodes = node.getChildNodes();
133
134         for (int j = 0; j < childNodes.getLength(); j++) {
135
136             Node childNode = childNodes.item(j);
137
138             if ("extra".equals(childNode.getNodeName())) {
139
140                 if (Integer.valueOf(childNode.getTextContent()) <= 20000) {
141
142                     childNode.setTextContent(textContent:"15000");
143                 }
144             }
145         }
146     }
147 }
148
149
150 //a modositott adatok kiirasa konzolra
151
152 TransformerFactory transformerFactory = TransformerFactory.newInstance();
153 Transformer transformer = transformerFactory.newTransformer();
154
155 DOMSource source = new DOMSource(document);
156
157 System.out.println(x:"---- Modositott fajl -----");
158
159 StreamResult consoleResult = new StreamResult(System.out);
160
161 transformer.transform(source, consoleResult);
162
163
164 } catch (Exception e ) {
165     e.printStackTrace();
166 }
167
168
169
170
171 }
172

```


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 autóiinak kiírása amelyek 2020 vagy attól kisebb évjáratúak.
4. Írjuk ki a 'Git Áron' elérhetőségeit
5. A 2-es telep alkalmazottjainak az adatai

```
1  package hu.domparsing.h2z4x3;
2
3  import java.io.File;
4  import java.io.IOException;
5
6  import javax.xml.parsers.DocumentBuilder;
7  import javax.xml.parsers.DocumentBuilderFactory;
8  import javax.xml.parsers.ParserConfigurationException;
9
10 import org.w3c.dom.Document;
11 import org.w3c.dom.Element;
12 import org.w3c.dom.NamedNodeMap;
13 import org.w3c.dom.Node;
14 import org.w3c.dom.NodeList;
15 import org.xml.sax.SAXException;
16
17 public class DomQueryH2Z4X3 {
18
19     Run | Debug
20     public static void main(String[] args) throws IOException, ParserConfigurationException, SAXException {
21         try {
22
23             //xml file megnyitása adatok beolvasásához
24             File file = new File(pathname:"XMLh2z4x3.xml");
25
26             //dokumentum létrehozása a beolvasott file-ból
27             DocumentBuilder documentBuilder = DocumentBuilderFactory.newInstance().newDocumentBuilder();
28             Document document = documentBuilder.parse(file);
29
30             //dokumentum normalizálása
31             document.getDocumentElement().normalize();
32
33
34             lekerdezes1(document);
35             lekerdezes2(document);
36             lekerdezes3(document);
37             lekerdezes4(document);
38             lekerdezes5(document);
39             lekerdezes6(document);
40
41
42             } catch (IOException e) {
43                 e.printStackTrace();
44             } catch (ParserConfigurationException e) {
45                 e.printStackTrace();
46             } catch (SAXException e) {
47                 e.printStackTrace();
48             }
49         }
50     }
51 }
```

```

51
52 public static void lekerdezes1(Document document) {
53
54     try {
55
56         System.out.println(x:"\n1. Az A minősítésű telepek kapacitásának lekérdezése");
57         NodeList nList1 = document.getElementsByTagName(tagname:"telep");
58
59         for (int i = 0; i < nList1.getLength(); i++) {
60
61             Node nNode = nList1.item(i);
62
63             if (nNode.getNodeType()==Node.ELEMENT_NODE) {
64
65                 Element elem = (Element) nNode;
66                 NodeList childNodes = nNode.getChildNodes();
67
68                 for (int j = 0; j < childNodes.getLength(); j++) {
69
70                     Node childNode = childNodes.item(j);
71
72                     if ("minősítés".equals(childNode.getNodeName())) {
73
74                         if ("A".equals(childNode.getTextContent())) {
75                             System.out.println(x:"");
76
77                             String telepid = elem.getAttribute(name:"TELEP_ID");
78                             Node node1 = elem.getElementsByTagName(name:"kapacitás").item(index:0);
79                             String node1Text = node1.getTextContent();
80
81
82                             System.out.println("Telep_ID: " + telepid);
83                             System.out.println("kapacitás: " + node1Text);
84
85
86                         }
87                     }
88                 }
89             }
90
91         }
92         System.out.println(x:"*****");
93
94     } catch (NullPointerException e) {
95         e.printStackTrace();
96     }
97
98 ~
99 public static void lekerdezes2(Document document) {
100
101     try {
102
103         System.out.println(x:"\n2. Az 1500000ft-nál nagyobb bevételű telep tulajdonosának kiírása");
104         NodeList nList2 = document.getElementsByTagName(tagname:"tulajdonos");
105
106         for (int i = 0; i < nList2.getLength(); i++) {
107
108             Node nNode = nList2.item(i);
109
110             if (nNode.getNodeType()==Node.ELEMENT_NODE) {
111
112                 Element elem = (Element) nNode;
113                 NodeList childNodes = nNode.getChildNodes();
114
115                 NamedNodeMap attr = nNode.getAttributes();
116
117
118                 if (nNode.getNodeType()==Node.ELEMENT_NODE) {
119                     for (int j = 0; j < childNodes.getLength(); j++) {
120
121                         Node childNode = childNodes.item(j);
122
123                         if ("bevétel".equals(childNode.getNodeName())) {
124
125                             if (Integer.valueOf(childNode.getTextContent()) > 1500000 ) {
126                                 System.out.println(x:"");
127
128                                 String tulajdonosid = elem.getAttribute(name:"TUL_ID");
129                                 Node node1 = elem.getElementsByTagName(name:"név").item(index:0);
130                                 String node1Text = node1.getTextContent();
131                                 System.out.println("Tulajdonos_ID: " + tulajdonosid);
132                                 System.out.println("név: " + node1Text);
133
134                             }
135                         }
136                     }
137                 }
138             }
139
140         }
141
142         System.out.println(x:"*****");
143
144     } catch (NullPointerException e) {
145         e.printStackTrace();
146     }
147
148 }
149
150 }
151
152 ~

```

```

153     }
154
155     public static void lekerdezes3(Document document) {
156
157         try {
158
159             System.out.println(x:"\n3.: A '2'-es ID-jű telep autóinak kiíratása amelyek 2020 vagy attól kisebb évjártúak.");
160             NodeList nList3 = document.getElementsByTagName(tagname:"autok");
161
162             for (int i = 0; i < nList3.getLength(); i++) {
163
164                 Node nNode = nList3.item(i);
165
166                 if (nNode.getNodeType() != Node.ELEMENT_NODE) {
167
168                     Element elem = (Element) nNode;
169                     NodeList childNodes = nNode.getChildNodes();
170
171
172                     NamedNodeMap attr = nNode.getAttributes();
173                     Node nodeAttribute = attr.getNamedItem(name:"AUTO_TELEP");
174
175                     if (nodeAttribute.getTextContent().equals(anObject:"2")) {
176                         for (int j = 0; j < childNodes.getLength(); j++) {
177
178                             Node childNode = childNodes.item(j);
179
180                             if ("gyártásiév".equals(childNode.getNodeName())) {
181
182                                 if (Integer.valueOf(childNode.getTextContent()) <= 2020 ) {
183                                     System.out.println(x:"");
184
185                                     String autoid = elem.getAttribute(name:"AUTO_ID");
186                                     Node node1 = elem.getElementsByTagName(name:"rendszer").item(index:0);
187                                     String node1Text = node1.getTextContent();
188                                     Node node2 = elem.getElementsByTagName(name:"modell").item(index:0);
189                                     String node2Text = node2.getTextContent();
190
191
192                                     System.out.println("Auto_ID: " + autoid);
193                                     System.out.println("Rendszám: " + node1Text);
194                                     System.out.println("Modell: " + node2Text);
195                                 }
196                             }
197                         }
198                     }
199                 }
200
201             }
202             System.out.println(x:"*****");
203
204
205         } catch (NullPointerException e) {
206             e.printStackTrace();
207         }
208     }
209
210 }
211

```

```

212 public static void lekendezes4(Document document) {
213
214     try {
215
216         System.out.println(x:"\n3.: A '2'-es ID-jű telep autóinak kiíratása amelyek 2020 vagy attól kisebb évjártatúak.");
217         NodeList nList3 = document.getElementsByTagName(tagname:"autok");
218
219         for (int i = 0; i < nList3.getLength(); i++) {
220
221             Node nNode = nList3.item(i);
222
223             if (nNode.getNodeType()==Node.ELEMENT_NODE) {
224
225                 Element elem = (Element) nNode;
226                 NodeList childNodes = nNode.getChildNodes();
227
228                 NamedNodeMap attr = nNode.getAttributes();
229                 Node nodeAttribute = attr.getNamedItem(name:"AUTO_TELEP");
230
231                 if (nodeAttribute.getTextContent().equals(anObject:"2")) {
232                     for (int j = 0; j < childNodes.getLength(); j++) {
233
234                         Node childNode = childNodes.item(j);
235
236                         if ("gyártásiév".equals(childNode.getNodeName())) {
237
238                             if (Integer.valueOf(childNode.getTextContent()) <= 2020 ) {
239                                 System.out.println(x:"");
240
241                                 String autoid = elem.getAttribute(name:"AUTO_ID");
242                                 Node node1 = elem.getElementsByTagName(name:"rendszám").item(index:0);
243                                 String node1Text = node1.getTextContent();
244                                 Node node2 = elem.getElementsByTagName(name:"modell").item(index:0);
245                                 String node2Text = node2.getTextContent();
246
247
248                                 System.out.println("Auto_ID: " + autoid);
249                                 System.out.println("Rendszám: " + node1Text);
250                                 System.out.println("modell: " + node2Text);
251                             }
252                         }
253                     }
254                 }
255             }
256         }
257
258         System.out.println(x:"*****");
259         System.out.println(x:"*****");
260
261
262     } catch (NullPointerException e) {
263         e.printStackTrace();
264     }
265 }
266
267
268
269 public static void lekendezes5(Document document) {
270
271     try {
272         System.out.println(x:"\n5. Írjuk ki a 'Git Áron' elérhetőségeit");
273         NodeList nList5 = document.getElementsByTagName(tagname:"bérlő");
274
275         for (int i = 0; i < nList5.getLength(); i++) {
276
277             Node nNode = nList5.item(i);
278
279             if (nNode.getNodeType()==Node.ELEMENT_NODE) {
280
281                 Element elem = (Element) nNode;
282                 NodeList childNodes = nNode.getChildNodes();
283
284                 for (int j = 0; j < childNodes.getLength(); j++) {
285
286                     Node childNode = childNodes.item(j);
287
288                     if ("név".equals(childNode.getNodeName())) {
289
290                         if ("Git Áron".equals(childNode.getTextContent())) {
291                             System.out.println(x:"");
292
293                             String telepid = elem.getAttribute(name:"BÉRLŐ_ID");
294                             Node node1 = elem.getElementsByTagName(name:"telszám").item(index:0);
295                             String node1Text = node1.getTextContent();
296                             Node node2 = elem.getElementsByTagName(name:"email").item(index:0);
297                             String node2Text = node2.getTextContent();
298
299                             System.out.println("Bérlő_id: " + telepid);
300                             System.out.println("telszám: " + node1Text);
301                             System.out.println("email: " + node2Text);
302                         }
303                     }
304                 }
305             }
306         }
307
308         System.out.println(x:"*****");
309
310     } catch (NullPointerException e) {
311         e.printStackTrace();
312     }
313 }
314
315
316

```

```

319
320 public static void lekerdezes6(Document document) {
321
322     try {
323         System.out.println(x:"\n6. A 2-es telep alkalmazottianak az adatai");
324         NodeList nList6 = document.getElementsByTagName(tagname:"alkalmazott");
325
326         for (int i = 0; i < nList6.getLength(); i++) {
327
328             Node nNode = nList6.item(i);
329
330             if (nNode.getNodeType()==Node.ELEMENT_NODE) {
331
332                 Element elem = (Element) nNode;
333                 NamedNodeMap attr = nNode.getAttributes();
334                 Node nodeAttrM = attr.getNamedItem(name:"ALK_TELEP");
335
336                 if (nodeAttrM.getTextContent().equals(anObject:"2")) {
337
338                     System.out.println(x:"");
339
340                     String alkalmazottid = elem.getAttribute(name:"ALK_ID");
341                     Node node1 = elem.getElementsByTagName(name:"név").item(index:0);
342                     String node1Text = node1.getTextContent();
343                     Node node2 = elem.getElementsByTagName(name:"beosztás").item(index:0);
344                     String node2Text = node2.getTextContent();
345                     Node node3 = elem.getElementsByTagName(name:"életkor").item(index:0);
346                     String node3Text = node3.getTextContent();
347                     Node node4 = elem.getElementsByTagName(name:"születésiév").item(index:0);
348                     String node4Text = node4.getTextContent();
349
350
351                     System.out.println("Alkalmazott_ID: " + alkalmazottid);
352                     System.out.println("Név: " + node1Text);
353                     System.out.println("Beosztás: " + node2Text);
354                     System.out.println("Életkor: " + node3Text);
355                     System.out.println("Születési év: " + node4Text);
356                 }
357             }
358         }
359
360         System.out.println(x:"*****");
361
362     } catch (NullPointerException e) {
363         e.printStackTrace();
364     }
365 }
366
367
368
369

```

2d) adatírás

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

```
1 package hu.domparse.h2z4x3;
2
3 import java.io.File;
4 import java.io.FileOutputStream;
5 import java.io.IOException;
6 import java.io.OutputStream;
7 import javax.xml.parsers.DocumentBuilder;
8 import javax.xml.parsers.DocumentBuilderFactory;
9 import javax.xml.parsers.ParserConfigurationException;
10 import javax.xml.transform.OutputKeys;
11 import javax.xml.transform.Transformer;
12 import javax.xml.transform.TransformerConfigurationException;
13 import javax.xml.transform.TransformerException;
14 import javax.xml.transform.TransformerFactory;
15 import javax.xml.transform.dom.DOMSource;
16 import javax.xml.transform.stream.StreamResult;
17 import org.w3c.dom.Document;
18 import org.w3c.dom.Node;
19 import org.w3c.dom.NodeList;
20
21 public class DomWriteH2Z4X3 {
22
23
24     public static void main(String[] args)
25     {
26
27         try
28         {
29             File inputFile = new File(pathname:"XMLh2z4x3.xml");
30             DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
31             DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
32             Document doc = dBuilder.parse(inputFile);
33             doc.getDocumentElement().normalize();
34
35
36             //Fa struktúra
37             printNode(doc.getDocumentElement(), depth:0);
38
39             TransformerFactory transformerFactory = TransformerFactory.newInstance();
40             Transformer transformer = transformerFactory.newTransformer();
41             transformer.setOutputProperty(OutputKeys.INDENT, value:"yes"); // Bekezdések hozzáadása
42             transformer.setOutputProperty(name:"{http://xml.apache.org/xslt}indent-amount", value:"4"); // Indentálási mélység
43
44             DOMSource source = new DOMSource(doc);
45             StreamResult result = new StreamResult(new File(pathname:"XMLh2z4x3_1.xml"));
46             transformer.transform(source, result);
47
48         }
49         catch (Exception e)
50         {
51             e.printStackTrace();
52         }
53     }
54
55     private static void printNode(Node node, int depth)
56     {
57
58         String indent = " ".repeat(depth * 4);
59
60         if (node.getNodeType() == Node.ELEMENT_NODE)
61         {
62
63             System.out.println(indent + "<" + node.getNodeName() + ">");
64             NodeList nodeList = node.getChildNodes();
65
66             for (int i = 0; i < nodeList.getLength(); i++)
67             {
68                 printNode(nodeList.item(i), depth + 1);
69             }
70             System.out.println(indent + "</" + node.getNodeName() + ">");
71         }
72         else if (node.getNodeType() == Node.TEXT_NODE && !node.getTextContent().trim().isEmpty())
73         {
74             System.out.println(indent + node.getTextContent().trim());
75         }
76     }
77
78 }
79 }
```