

# JEGYZŐKÖNYV

## Adatkezelés XML környezetben

Féléves feladat

Busz ágazat logisztikai rendszerezése

**Készítette:** Szepesi Glória Virág

**Neptunkód:** WOEORY

**Dátum:** 2024. 11. 19.

<b>1. A FELADAT LEÍRÁSA .....</b>	<b>3</b>
A FELADAT LÉPÉSEI:.....	3
AZ ADATBÁZIS ER MODELLJÉNEK TERVE:.....	5
AZ EGYEDEKRŐL RÉSZLETESEBBEN:.....	7
A KAPCSOLATOK:.....	8
AZ ADATBÁZIS XDM MODELLRE TÖRTÉNŐ ÁTALAKÍTÁSA .....	9
XML DOKUMENTUM LÉTREHOZÁSA AZ XDM MODELL ALAPJÁN .....	10
XML SCHEMA KÉSZÍTÉSE AZ XML DOKUMENTUMBÓL .....	15
 <b>2. FELADATRÉSZ.....</b>	<b>18</b>
DOM PROJECT .....	19
AZ ADATLEKÉRDEZÉS RÉSZE: .....	19
AZ ADATMÓDOSÍTÁS: .....	22
AZ ADATÍRÁS:.....	25

# 1. A feladat leírása

## Feladatléírás

A következő feladat célja egy autóbusz társaság adatainak adminisztrációja XML dokumentumok segítségével. A projekt keretében különböző lépések során szükséges megtervezni az adatbázis ER modelljét, átalakítani XDM modellé, majd az XDM modell alapján XML dokumentumot létrehozni. Végül az XML dokumentum alapján XMLSchema-t készítünk, és egy DOM program segítségével adminisztráljuk az adatokat.

## A feladat lépései

1. **ER modell tervezése:** Az autóbusz társaság adatainak modellezése entitások és kapcsolataik segítségével, például buszok, sofőrök, szervizek stb. meghatározása és az entitások közötti kapcsolatok definiálása.
2. **Az adatbázis XDM modellre történő átalakítása:** Az ER modell átalakítása XML Data Model (XDM) formátumra. Példa az XDM modell átalakítására:

```
<BusServiceSystem>

  <Szerviz>

    <SzID></SzID>

    <JavitóSzám></JavitóSzám>

    <Név></Név>

  </Szerviz>

  <Busz>

    <BID></BID>

    <Rendszam></Rendszam>

    <Tipus></Tipus>

    <max_hely></max_hely>

    <Szerviz>

      <SzID></SzID>

    </Szerviz>

  </Busz>

  ...

</BusServiceSystem>
```

3. **XML dokumentum létrehozása az XDM modell alapján:** Az XDM modell alapján egy XML dokumentum létrehozása. Például az XMLNeptunkod.xml fájl tartalma a következőképpen nézhet ki:

```
<Palyaudvar xsi:noNamespaceSchemaLocation="./XMLSchemaHR6121.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <Buszok>

    <Busz Id="001">

      <Rendszam>KWV-965</Rendszam>

      <Tipus>Mercedes</Tipus>

      <max_hely>52</max_hely>

    </Busz>

    <!-- További buszok -->

  </Buszok>

  <Soforok>

    <Sofor SoforId="101">

      <Jegykiadoszam>H37625</Jegykiadoszam>

      <Nev>Ragó Lili</Nev>

      <Busz Id_FK="007"></Busz>

    </Sofor>

    <!-- További sofőrök -->

  </Soforok>

  ...

</Palyaudvar>
```

4. **XMLSchema készítése az XML dokumentumból:** Az XML dokumentum alapján létrehozott séma az adatok szerkezetének és érvényességének ellenőrzésére szolgál.
5. **DOM program készítése az XML dokumentum adatai adminisztrálása alapján:** Egy Java program, amely betölti az XMLNeptunkod.xml fájlt, kiírja annak tartalmát fa struktúra formában a konzolra, és egy új fájlba XMLW0E0RY.xml néven menti.

## Az adatbázis ER modelljének terve

Az ER modellben nagyon fontos az átláthatóság. Ezt igyekeztem szem előtt tartani. Tanár úr is mindig kiemeli, hogy az összekötő elemek ne lógnak egymásba, valamint, hogy legyen szintenként elhelyezve minden.

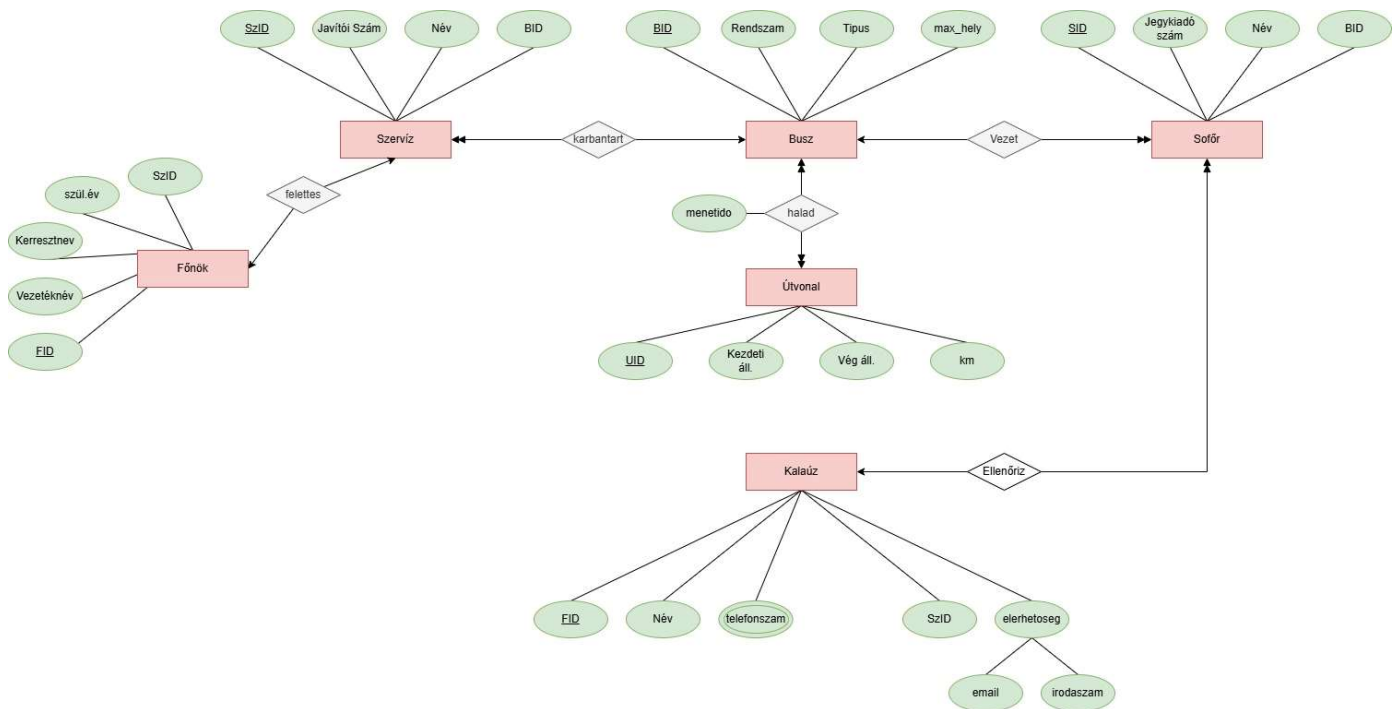
A modell tartalmaz minimum 5 egyedet, többféle kapcsolatot, fellelhető benne például 1:1 valamint M:N kapcsolat is. Az egyedek többféle tulajdonsággal rendelkeznek, van normál, id-azaz a kulcs- valamint többértékű is.

Az Az adatbázis egyedei, amelyeket a korábbi XML dokumentumban meghatároztunk, a következőképpen néznek ki:

- **Buszok**
  - Id: egyedi azonosító, numerikus
  - Rendszam: a jármű rendszáma
  - Tipus: a busz típusa
  - max\_hely: numerikus szám
- **Sofőrök**
  - SoforID: a Sofőr egyedi kulcsa
  - Jegykiadószám:
  - Név: A sofőr teljes neve
  - BuszokID: egy idegen kulcs, ami a busz egyedhez kötődik
- **Szervizesek**
  - SzervizID: egyedi azonosító, numerikus
  - Javítószám: azonosítószám. numerikus
  - Név: a Szerviz vagy szerelő neve
  - BuszokID: idegen kulcs, ami a busz egyedhez kötődik
- **Eszközök**
  - EszközID: egyedi azonosító, numerikus
  - Termék szám: A használt termék azonosítója. numerikus
  - Gyártó: a gyártó megnevezése, string
  - Év: Az eszköz gyártási éve
  - SzervízID: idegen kulcs, a szervízhez kapcsolódó azonosításhoz
- **Útvonalak**
  - UtvonalID: egyedi azonosító, numerikus
  - Kezd Áll: A kezdeti állomás ahonnan a Jármű kiindul
  - Vég Áll: A végállomás ahová a jármű megérkezik
  - Km: Az utazás során megtett kilométer
- **Halad:**
  - UtvonalID: Idegen kulcs
  - BuszokID: Idegen kulcs
  - Menetido: Megadja a megtett utat az idő tekintetében

- **Kalauzok**

- KalauzID: egyéni azonosító, numerikus
- Név: a Kalauz/ellenőr teljes neve
- SoforID: idegen kulcs a sofőrök meghatározására
- SzervizID: idegen kulcs, a szerviz meghatározásához
- Elérhetőség (Irodaszám, Email): A személyek elérhetősége.
- Telefonszám: Az érintett kalauz telefonszáma, numerikus.



## Az Egyedekről részletesebben

### A Busz egyed:

Ez az egyed egy központibb egyed. Itt találhatóak meg az buszok alap adatai. az ID az egyed elsődleges kulcsa. Emellett megadható még a busz rendszáma, a gyárója ez string típusú tulajdonság mindkettő. Jellemezhetjük még a buszt a férőhelyével, hogy hány utast szállíthat egyszerre, ami integer típusú.

### A Szerviz egyed:

A szervíz egyed a buszokhoz tartozó szerviz adatait tárolja. Rendelkezik egy „k\_id” azonosítóval, ami itt az elsődleges kulcs. Van még neve, mint string adat, egy azonosító, ami integer típusú. Van még itt egy Foreign key, ami a buszhoz kapcsolja az egyedet.

### A Főnök egyed:

A Főnök egyed egy 1:1 kapcsolattal kapcsolódik a szervízhez, minden szervíznek/ ott dolgozónak van egy felettese. Ezekhez tartozik egy elsődleges kulcs (id) egy vezetéknév és keresztnév, ami szöveges adatként szerepel. Mindenkihez tartozik egy születési dátum, valamint összeköttetés miatt egy idegenkulcsként itt is fellelhető egy szerviz\_id.

### A Sofőr egyed:

Ebben az egyedben találjuk meg a buszokat vezető emberek adatait. Találunk egy SID – a sofőrök azonosítója- itt ez az elsődleges kulcs. A Vezetők egyértelműen rendelkeznek névvel, ami szöveges bevitelt vár, egy- egy jegykiadó számmal, ami integer vár. A megjelenő idegen kulcs pedig a buszok egyedi azonosítói.

### Az útvonal egyed:

Ez az egyed tartalmazza a sofőrök által levezetett útvonalat. Az id jelölésű rész itt is az elsődleges kulcsot jelenti, Van kezdet- és végállomás, ahova szöveges adatként tudjuk beírni a településeket/ megállókat. Tárol még az egyed egy kilométer számot, ami a két állomás közötti távolságot jelenti.

### A Kalauz egyed:

Végül de nem utolsó sorban a Kalauz egyed. Ez az egyed, aki ellenőrzi a buszokat, utasokat, valamint a sofőröket. Van egy ID, ami az elsődleges kulcs. Rendelkezik egy névvel, ami szöveges bevitelt igényel. Találunk még egy összetett tulajdonságot, ami az elérhetőség. Ezen belül van a szöveges típusú email cím, valamint egy irodaszám, amit számokkal tudunk megadni.

Eltárol még egy telefonszám tulajdonságot, ami akár több adatot is tud rögzíteni tehát ez többértékű tulajdonság. Csatlakozik még ehhez két idegen kulcs.

## A Kapcsolatok

### A Szervíz és a Főnök kapcsolata:

E kettő egyed közötti kapcsolat egy-egy kapcsolat, mivel egy szervízhez esetleg szerelőhöz egy felettest rendelünk.

### A Szervíz és Busz kapcsolata:

Itt a kapcsolat egy a többhöz kapcsolat, ugyanis egy szervíz több műszakilag károsult járművet is tud szervízeltetni, ugyanez szerelőkre értve nyilván.

### A Busz és a Sofőr kapcsolat:

Itt is egy a több kapcsolat áll fenn a két egyed között ugyanis egy járművet nyilván egy személy vezet, de például egy hosszú út során a busz rendelkezhet több sofőrrel is.

### A Busz és Útvonal kapcsolat:

Ez a kapcsolat több a többhöz azaz M:N kapcsolat. Egy jármű több útvonalon is közlekedhet, valamint egy útvonalat több busz is bejárhat. Ide köthetünk még egy „menetidő” tulajdonságot.

### A Kalauz és a Sofőr kapcsolata:

Itt is egy a többhöz kapcsolat áll fenn, ugyanis egy kalauz több sofőrt is ellenőrizhet, ám egy a volán mögött ülő személyt egyszerre csak egy ember tud ellenőrizni.



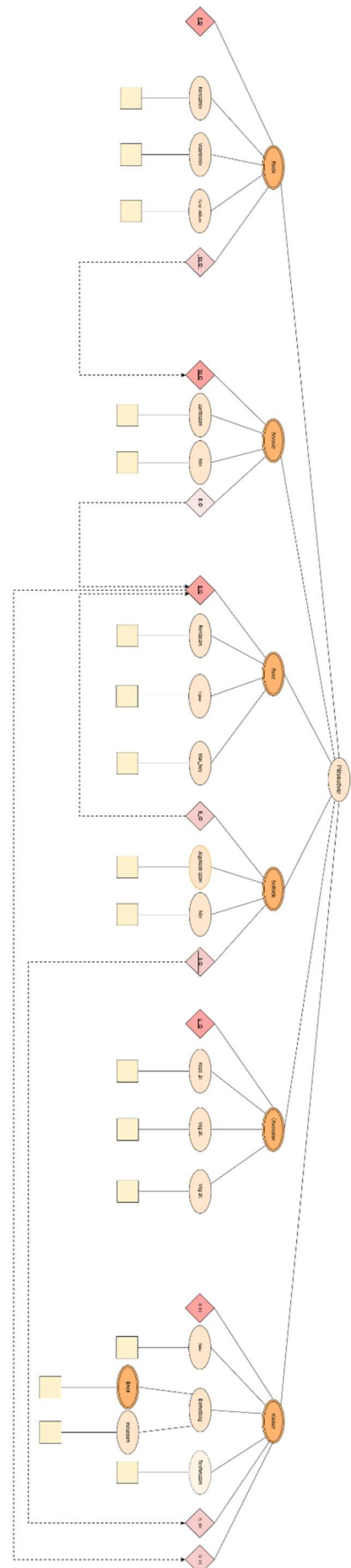
## Az adatbázis XDM modellre történő átalakítása

Az Experience Data Model (XDM) egy olyan adatmodell, amelyet az Adobe fejlesztett ki annak érdekében, hogy a vállalkozások jobban integrálhassák, kezelhessék és elemezhesék az ügyfeladatokat. Az XDM célja, hogy egységesítsen különböző adatforrásokat és struktúrákat, ezáltal könnyebbé téve az adatok kezelését és felhasználását az ügyfélművelés javítása érdekében.

Az első feladat b, része az XDM lemodellezése volt. Ennek alapjául szolgált az előbb bemutatott ER modell. Figyelembe kell venni a példányosítás folyamatát, hogy átlátható és könnyen lekövethető legyen az ábra. Mindent egy szülő egyedbe vezettem vissza a többszörösíték miatt.

A folyamat során szükséges volt létrehozni egy új osztályt, ami a Halad névre hallgat. Ezt azért kellett létrehozni, mert található az ER modellben egy több a többhöz kapcsolat, ennek az átképzése.

Az összekötő gyökérelem a pályaudvar lett.



## XML dokumentum létrehozása az XDM modell alapján

Az XML dokumentumot az XDM modell alapján készítettem. A gyökérelem a pályaudvar lett.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <Palyaudvar xsi:noNamespaceSchemaLocation="./XMLSchemaHR6121.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
4
5      <!--Autóbuszok példányosítása-->
6
7      <Buszok>
8
9          <Buszok Id="001">
10              <Rendszam> KKV-965 </Rendszam>
11              <Tipus> Mercedes </Tipus>
12              <max_hely> 52 </max_hely>
13          </Buszok>
14
15          <Buszok Id="002">
16              <Rendszam> KKH-040 </Rendszam>
17              <Tipus> Ikarus </Tipus>
18              <max_hely> 40 </max_hely>
19          </Buszok>
20
21          <Buszok Id="003">
22              <Rendszam> PTE-102 </Rendszam>
23              <Tipus> Ikarus </Tipus>
24              <max_hely> 40 </max_hely>
25          </Buszok>
26
27          <Buszok Id="004">
28              <Rendszam> GEC-420 </Rendszam>
29              <Tipus> MAN </Tipus>
30              <max_hely> 50 </max_hely>
31          </Buszok>
32
33          <Buszok Id="005">
34              <Rendszam> KKK-012 </Rendszam>
35              <Tipus> Mercedes </Tipus>
36              <max_hely> 52 </max_hely>
37          </Buszok>
38
39          <Buszok Id="006">
40              <Rendszam> GDP-456 </Rendszam>
41              <Tipus> MAN </Tipus>
42              <max_hely> 50 </max_hely>
43          </Buszok>
44
45          <Buszok Id="007">
46              <Rendszam> LPG-324 </Rendszam>
47              <Tipus> Ikarus </Tipus>
48              <max_hely> 40 </max_hely>
49          </Buszok>
50
51          <Buszok Id="008">
52              <Rendszam> KRV-666 </Rendszam>
53              <Tipus> MAN </Tipus>
54              <max_hely> 50 </max_hely>
55          </Buszok>
56
57          <Buszok Id="009">
58              <Rendszam> RPL-104 </Rendszam>
59              <Tipus> Mercedes </Tipus>
60              <max_hely> 52 </max_hely>
61          </Buszok>
62
63      <Buszok ok>
64
65      <Soforok>
66
67          <Sofor SoforId="101">
68              <Jegykiadoszam> H37625 </Jegykiadoszam>
69              <Nev> Ragó Lili </Nev>
70              <Buszok Id_FK> 007 <Buszok Id_FK>
71          </Sofor>
72
73          <Sofor SoforId="102">
74              <Jegykiadoszam> K47382 </Jegykiadoszam>
75              <Nev> Zakar Edit Dóra </Nev>
76              <Buszok Id_FK> 001 <Buszok Id_FK>
77          </Sofor>
78
79          <Sofor SoforId="103">
80              <Jegykiadoszam> L98237 </Jegykiadoszam>
81              <Nev> Andó Viktória </Nev>
82              <Buszok Id_FK> 006 <Buszok Id_FK>
83          </Sofor>
84
85          <Sofor SoforId="104">
```

```

81         <Nev> Andó Viktória </Nev>
82         <Buszok Id_FK> 006 <Buszok Id_FK>
83     </Sofor>
84
85     <Sofor SoforId="104">
86         <Jegykiadoszam> B25869 </Jegykiadoszam>
87         <Nev> Szabó Attila </Nev>
88         <Buszok Id_FK> 002 <Buszok Id_FK>
89     </Sofor>
90
91     <Sofor SoforId="105">
92         <Jegykiadoszam> S14785 </Jegykiadoszam>
93         <Nev> Molnár Lili Noémi </Nev>
94         <Buszok Id_FK> 005 <Buszok Id_FK>
95     </Sofor>
96
97     <Sofor SoforId="106">
98         <Jegykiadoszam> 013579 </Jegykiadoszam>
99         <Nev> Soltész Bianka Gizella </Nev>
00         <Buszok Id_FK> 004 <Buszok Id_FK>
01     </Sofor>
02
03     <Sofor SoforId="107">
04         <Jegykiadoszam> U45691 </Jegykiadoszam>
05         <Nev> Rása Valentia Izabella </Nev>
06         <Buszok Id_FK> 009 <Buszok Id_FK>
07     </Sofor>
08
09     <Sofor SoforId="108">
10         <Jegykiadoszam> W11335 </Jegykiadoszam>
11         <Nev> Takács Viktória </Nev>
12         <Buszok Id_FK> 008 <Buszok Id_FK>
13     </Sofor>
14
15     <Sofor SoforId="109">
16         <Jegykiadoszam> J43434 </Jegykiadoszam>
17         <Nev> Ternay Frank </Nev>
18         <Buszok Id_FK> 003 <Buszok Id_FK>
19     </Sofor>
20
21 </Soforok>
22
23 <Szervizek>
24
25     <Szerviz SzervizId="201">
26         <Javitoszam> C6005 </Javitoszam>
27         <Nev> Bihari Panna </Nev>
28         <Buszok Id_FK> 006 <Buszok Id_FK>
29     </Szerviz>
30
31     <Szerviz SzervizId="202">
32         <Javitoszam> M8009 </Javitoszam>
33         <Nev> Ragó Lajos </Nev>
34         <Buszok Id_FK> 007 <Buszok Id_FK>
35     </Szerviz>
36
37     <Szerviz SzervizId="203">
38         <Javitoszam> Z7003 </Javitoszam>
39         <Nev> Pál Vajk </Nev>
40         <Buszok Id_FK> 004 <Buszok Id_FK>
41     </Szerviz>
42
43     <Szerviz SzervizId="204">
44         <Javitoszam> X5114 </Javitoszam>
45         <Nev> Náray Áron </Nev>
46         <Buszok Id_FK> 002 <Buszok Id_FK>
47     </Szerviz>
48
49     <Szerviz SzervizId="205">
50         <Javitoszam> E8877 </Javitoszam>
51         <Nev> Nagy Tamás Roland </Nev>
52         <Buszok Id_FK> 001 <Buszok Id_FK>
53     </Szerviz>
54 </Szervizek>
55
56 <Eszkozok>
57     <Eszkoz EszkozId="301">
58         <Termekszam> 97 </Termekszam>
59         <Gyarto> Bosch </Gyarto>
60         <Ev> 1 </Ev>
61         <SzervizId_FK> 201 </SzervizId_FK>
62     </Eszkoz>
63

```

```

163
164     <Eszkoz EszkozId="302">
165         <Termekszam> 42 </Termekszam>
166         <Gyarto> Bosch </Gyarto>
167         <Ev> 3 </Ev>
168         <SzervizId_FK> 202 </SzervizId_FK>
169     </Eszkoz>
170
171     <Eszkoz EszkozId="303">
172         <Termekszam> 55 </Termekszam>
173         <Gyarto> Bosch </Gyarto>
174         <Ev> 5 </Ev>
175         <SzervizId_FK> 203 </SzervizId_FK>
176     </Eszkoz>
177
178     <Eszkoz EszkozId="304">
179         <Termekszam> 12 </Termekszam>
180         <Gyarto> Bosch </Gyarto>
181         <Ev> 2 </Ev>
182         <SzervizId_FK> 204 </SzervizId_FK>
183     </Eszkoz>
184
185     <Eszkoz EszkozId="305">
186         <Termekszam> 08 </Termekszam>
187         <Gyarto> Bosch </Gyarto>
188         <Ev> 7 </Ev>
189         <SzervizId_FK> 205 </SzervizId_FK>
190     </Eszkoz>
191
192 </Eszkozok>
193
194 <Utvonalak>
195     <Utvonal UtvonalId="401">
196         <KezdAll> Felső Majláth </KezdAll>
197         <VegAll> Tiszai pu. </VegAll>
198         <Km> 12 </Km>
199     </Utvonal>
200
201     <Utvonal UtvonalId="402">
202         <KezdAll> Majális park </KezdAll>
203         <VegAll> Népkert </VegAll>
204         <km> 7 </km>
205     </Utvonal>
206
207     <Utvonal UtvonalId="403">
208         <KezdAll> Újgyőri főtérr </KezdAll>
209         <VegAll> Ávasi Kilátó </VegAll>
210         <km> 9 </km>
211     </Utvonal>
212
213     <Utvonal UtvonalId="404">
214         <KezdAll> Vasgyár </KezdAll>
215         <VegAll> Centrum </VegAll>
216         <km> 7 </km>
217     </Utvonal>
218
219     <Utvonal UtvonalId="405">
220         <KezdAll> Repülőtér </KezdAll>
221         <VegAll> Bábonyi bér </VegAll>
222         <km> 10 </km>
223     </Utvonal>
224
225 </Utvonalak>
226
227 <Haladok>
228     <Halad>
229         <UtvonalId_FK> 401 </UtvonalId_FK>
230         <Buszok ID_FK> 003 <Buszok ID_FK>
231         <Menetido> 25 </Menetido>
232     </Halad>
233
234     <Halad>
235         <UtvonalId_FK> 404 </UtvonalId_FK>
236         <Buszok ID_FK> 001 <Buszok ID_FK>
237         <Menetido> 26 </Menetido>
238     </Halad>
239
240     <Halad>
241         <UtvonalId_FK> 405 </UtvonalId_FK>
242         <Buszok ID_FK> 007 <Buszok ID_FK>
243         <Menetido> 19 </Menetido>
244     </Halad>

```

```

245
246         <Halad>
247             <UtvonalId_FK> 402 </UtvonalId_FK>
248             <Buszok ID_FK> 002 <Buszok ID_FK>
249             <Menetido> 24 </Menetido>
250         </Halad>
251
252         <Halad>
253             <UtvonalId_FK> 404 </UtvonalId_FK>
254             <Buszok ID_FK> 003 <Buszok ID_FK>
255             <Menetido> 21 </Menetido>
256         </Halad>
257
258         <Halad>
259             <UtvonalId_FK> 403 </UtvonalId_FK>
260             <Buszok ID_FK> 009 <Buszok ID_FK>
261             <Menetido> 20 </Menetido>
262         </Halad>
263
264         <Halad>
265             <UtvonalId_FK>406 </UtvonalId_FK>
266             <Buszok ID_FK> 001 <Buszok ID_FK>
267             <Menetido> 23 </Menetido>
268         </Halad>
269
270     </Haladok>
271
272     <Kalauz>
273         <Kalauz KalauzId="501">
274             <Nev> Pataki Csanad </Nev>
275             <SoforId_FK> 102 </SoforId_FK>
276             <SzervizId_FK> 204 </SzervizId_FK>
277             <Elerhetoseg>
278                 <Irodaszam> 11 </Irodaszam>
279                 <Email> pat.csan@gmail.com </Email>
280             </Elerhetoseg>
281             <Telefonszam> 06 55 584 888 </Telefonszam>
282             <Telefonszam> +36 55 563 979 </Telefonszam>
283         </Kalauz>
284
285         <Kalauz KalauzId="502">
286             <Nev> Gabor Borbala </Nev>
287             <SoforId_FK> 104 </SoforId_FK>
288             <SzervizId_FK> 202 </SzervizId_FK>
289             <Elerhetoseg>
290                 <Irodaszam> 12 </Irodaszam>
291                 <Email> gab.bor@gmail.com </Email>
292             </Elerhetoseg>
293             <Telefonszam> 06 55 010 226 </Telefonszam>
294             <Telefonszam> +36 55 686 186 </Telefonszam>
295         </Kalauz>
296
297         <Kalauz KalauzId="503">
298             <Nev> Szabó Oszkár </Nev>
299             <SoforId_FK> 103 </SoforId_FK>
300             <SzervizId_FK> 201 </SzervizId_FK>
301             <Elerhetoseg>
302                 <Irodaszam> 13 </Irodaszam>
303                 <Email> szab.osz@gmail.com </Email>
304             </Elerhetoseg>
305             <Telefonszam> 06 55 686 186 </Telefonszam>
306             <Telefonszam> +36 55 010 226 </Telefonszam>
307         </Kalauz>
308
309         <Kalauz KalauzId="504">
310             <Nev> Hegyi Borbala </Nev>
311             <SoforId_FK> 108 </SoforId_FK>
312             <SzervizId_FK> 203 </SzervizId_FK>
313             <Elerhetoseg>
314                 <Irodaszam> 14 </Irodaszam>
315                 <Email> hegy.bor@gmail.com </Email>
316             </Elerhetoseg>
317             <Telefonszam> 06 55 563 979 </Telefonszam>
318             <Telefonszam> +36 55 584 888 </Telefonszam>
319         </Kalauz>
320

```

```

321      <Kalauz KalauzId="505">
322          <Nev> Nagy Andrea </Nev>
323          <SoforId_FK> 109 </SoforId_FK>
324          <SzervizId_FK> 205 </SzervizId_FK>
325          <Elerhetoseg>
326              <Irodaszam> 15 </Irodaszam>
327              <Email> nagy.and@gmail.com </Email>
328          </Elerhetoseg>
329          <Telefonszam> 06 55 923 742 </Telefonszam>
330          <Telefonszam> +36 55 923 742 </Telefonszam>
331      </Kalauz>
332
333  </Kalauz>
334
335  </Palyaudvar>

```

## XMLSchema készítése az XML dokumentumból

Ebben a részben minden egyes egyedhez egy-egy saját változót csináltam, majd feltöltöttem példákkal.

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

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">

  <xs:element name="Palyaudvar">
    <xs:complexType>
      <xs:sequence>

        <xs:element name="Fonokok">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Fonok" type="FonokTip" minOccurs="0" maxOccurs="unbounded" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>

        <xs:element name="Szerviz">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="szerviz" type="szervizTip" minOccurs="0" maxOccurs="unbounded" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>

        <xs:element name="Buszok">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Busz" type="BuszTip" minOccurs="0" maxOccurs="unbounded" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>

        <xs:element name="Soforok">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Sofor" type="SoforTip" minOccurs="0" maxOccurs="unbounded" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>

        <xs:element name="Utvonalak">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Utvonal" type="UtvonalTip" minOccurs="0" maxOccurs="unbounded" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>

        <xs:element name="Kalauzok">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Kalauz" type="KalauzTip" minOccurs="0" maxOccurs="unbounded" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>

        <xs:element name="Haladok">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Halad" type="HaladTip" minOccurs="0" maxOccurs="unbounded" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>

      </xs:sequence>
    </xs:complexType>

    <xs:unique name="FonokId_FK">
      <xs:selector xpath="Fonokok/Fonok"/>
      <xs:field xpath="@FonokId"/>
    </xs:unique>

    <xs:keyref name="BuszID_FK-szerviz" refer="BuszID_FK">
      <xs:selector xpath="Szerviz/szerviz/BuszId_FK"/>
      <xs:field xpath="."/>
    </xs:keyref>
  </xs:element>
</xs:schema>
```



```

        <xs:keyref name="BuszID_FK-Sofor" refer="BuszID_FK">
            <xs:selector xpath="Soforok/Sofor/BuszId_FK"/>
            <xs:field xpath="."/>
        </xs:keyref>

        <xs:keyref name="BuszID_FK-Halad" refer="BuszID_FK">
            <xs:selector xpath="Haladas/Halad/BuszId_FK"/>
            <xs:field xpath="."/>
        </xs:keyref>

        <xs:keyref name="szervizID_FK-Kalauz" refer="szervizID_FK">
            <xs:selector xpath="Kalauzk/Kalauz/szervizId_FK"/>
            <xs:field xpath="."/>
        </xs:keyref>

        <xs:keyref name="SoforID_FK-Kalauz" refer="SoforID_FK">
            <xs:selector xpath="Kalauzk/Kalauz/SoforId_FK"/>
            <xs:field xpath="."/>
        </xs:keyref>
    </xs:element>

    <xs:complexType name="FonokTipus">
        <xs:sequence>
            <xs:element name="Gyarto" type="xs:string"/>
            <xs:element name="HasznaltEvek" type="xs:integer"/>
            <xs:element name="TermekSzam" type="xs:integer"/>
            <xs:element name="szervizId_FK" type="xs:integer"/>
        </xs:sequence>
        <xs:attribute name="FonokId" use="required">
            <xs:simpleType>
                <xs:restriction base="xs:integer">
                    <xs:pattern value="[0-9][0-9][0-9]" />
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>

    <xs:complexType name="szervizTipus">
        <xs:sequence>
            <xs:element name="Nev" type="xs:string"/>
            <xs:element name="JavitoiSzam" type="xs:integer"/>
            <xs:element name="BuszId_FK" type="xs:integer"/>
        </xs:sequence>
        <xs:attribute name="szervizId" use="required">
            <xs:simpleType>
                <xs:restriction base="xs:integer">
                    <xs:pattern value="[0-9][0-9][0-9]" />
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>

    <xs:complexType name="BuszTipus">
        <xs:sequence>
            <xs:element name="Max_hely" type="xs:integer"/>
            <xs:element name="Tipus" type="xs:string"/>
            <xs:element name="Rendszam" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="BuszId" use="required">
            <xs:simpleType>
                <xs:restriction base="xs:integer">
                    <xs:pattern value="[0-9][0-9][0-9]" />
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>

    <xs:complexType name="SoforTipus">
        <xs:sequence>
            <xs:element name="JegykiadoSzam" type="xs:integer"/>
            <xs:element name="Nev" type="xs:string"/>
            <xs:element name="TermekSzam" type="xs:string"/>
            <xs:element name="BuszId_FK" type="xs:integer"/>
        </xs:sequence>
        <xs:attribute name="SoforId" use="required">
            <xs:simpleType>
                <xs:restriction base="xs:integer">
                    <xs:pattern value="[0-9][0-9][0-9]" />
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>

```



```

<xs:complexType name="SoforTipus">
  <xs:sequence>
    <xs:element name="JegykiadoSzam" type="xs:integer"/>
    <xs:element name="Nev" type="xs:string"/>
    <xs:element name="TermekSzam" type="xs:string"/>
    <xs:element name="BuszId_FK" type="xs:integer"/>
  </xs:sequence>
  <xs:attribute name="SoforId" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:integer">
        <xs:pattern value="[0-9][0-9][0-9]" />
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>

<xs:complexType name="UtvonalTipus">
  <xs:sequence>
    <xs:element name="KezdetiAll" type="xs:string"/>
    <xs:element name="VegAll" type="xs:string"/>
    <xs:element name="km" type="xs:integer"/>
  </xs:sequence>
  <xs:attribute name="UtvonalId" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:integer">
        <xs:pattern value="[0-9][0-9][0-9]" />
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>

<xs:complexType name="KalauzTipus">
  <xs:sequence>
    <xs:element name="Nev" type="xs:String"/>
    <xs:element name="Telefonszam" type="xs:integer"/>
    <xs:element name="email" type="xs:string"/>
    <xs:element name="irodaszam" type="xs:integer"/>
    <xs:element name="SoforId_FK" type="xs:integer"/>
    <xs:element name="szervizId_FK" type="xs:integer"/>
  </xs:sequence>
  <xs:attribute name="szervizId" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:integer">
        <xs:pattern value="[0-9][0-9][0-9]" />
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>

<xs:complexType name="HaladTipus">
  <xs:sequence>
    <xs:element name="BuszId_FK" type="xs:integer"/>
    <xs:element name="UtvonalId_FK" type="xs:integer"/>
    <xs:element name="menetido" type="xs:integer"/>
  </xs:sequence>
</xs:complexType>

</xs:schema>

```

## 2. Feladatrész

```
1 package hudomparseW0e0ry;
2
3 import java.io.File;
4 import java.io.FileWriter;
5 import java.io.IOException;
6
7 import javax.xml.parsers.DocumentBuilder;
8 import javax.xml.parsers.DocumentBuilderFactory;
9 import javax.xml.parsers.ParserConfigurationException;
10
11 import org.w3c.dom.*;
12 import org.xml.sax.SAXException;
13
14 public class DomReadW0E0RY {
15
16     private static int baseLineRule = 0;
17     private final static String XMLFILE = "XMLW0E0RY.xml";
18     private static StringBuilder result;
19
20     public static void main(String[] args) throws IOException {
21         readOutAllElements();
22     }
23
24     //Eredmény kiírása
25     private static void resultToConsole() {
26         System.out.println(result.toString());
27     }
28
29     //Minden elem beolvasása
30     private static void readOutAllElements(){
31         Document doc;
32         DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
33         result = new StringBuilder();
34         try {
35             DocumentBuilder dBuilder = factory.newDocumentBuilder();
36             doc = dBuilder.parse(XMLFILE);
37
38             Node er = doc.getFirstChild();
39             NodeList erList = er.getChildNodes();
40             getAllElements(erList);
41         } catch (ParserConfigurationException | SAXException | IOException e) {
42             e.printStackTrace();
43         }
44
45         resultToConsole();
46     }
47
48     //Elemek összefűzése
49     private static void getAllElements(NodeList nl) {
50         Node node;
51         for (int i = 0; i < nl.getLength(); i++) {
52             node = nl.item(i);
53             if(node.getNodeType() == Node.ELEMENT_NODE) {
54
55                 result.append(indent(3)+node.getNodeName());
56
57                 if(node.hasAttributes()) {
58                     result.append(" "+ getAttributes(node.getAttributes()+"\n");//
59                 } else {
60                     result.append("\n");
61                 }
62                 if(node.getChildNodes().getLength() != 1) {
63
64                     NodeList childList = node.getChildNodes();
65                     baseLineRule++;
66                     getAllElements(childList);
67                 } else {
68                     baseLineRule++;
69                     result.append(indent(3)+node.getTextContent()+"\n");
70                     baseLineRule--;
71                 }
72             }
73         }
74         baseLineRule--;
```

## DOM project

### Az adatolvasás:

A fájlbeolvasás arról szól, hogy be tudjuk olvasni a korábban megírt dolgokat. Itt először is egy XML kiterjesztésű fájlt kell beolvasni.

```
74     baseLineRule--;
75 }
76
77 //Elemek elválasztása
78 private static String indent(int multiplyBy) {
79     StringBuilder whiteSpaces = new StringBuilder();
80     for (int i = 0; i < (baseLineRule *multiplyBy); i++) {
81         whiteSpaces.append(" ");
82     }
83     return whiteSpaces.toString();
84 }
85
86 //Elemek tulajdonságának a neve
87 private static String getAttributes(NamedNodeMap node) {
88     StringBuilder sb = new StringBuilder(" Attributes: [");
89     for (int i = 0; i < node.getLength(); i++) {
90         if(i != node.getLength()-1) {
91             sb.append(node.item(i).getNodeName()+" : "+node.item(i).getNodeValue()+" ");
92         } else {
93             sb.append(node.item(i).getNodeName()+" : "+node.item(i).getNodeValue());
94         }
95     }
96     sb.append("]");
97     return sb.toString();
98 }
99 }
```

### Az Adatlekérdezés

```

1  package hudomparseW0E0RY;
2
3  import java.io.File;
4  import java.io.IOException;
5
6  import javax.xml.parsers.DocumentBuilder;
7  import javax.xml.parsers.DocumentBuilderFactory;
8  import javax.xml.parsers.ParserConfigurationException;
9
10 import org.w3c.dom.Document;
11 import org.w3c.dom.Element;
12 import org.w3c.dom.Node;
13 import org.w3c.dom.NodeList;
14 import org.xml.sax.SAXException;
15
16 public class DomQueryW0E0RY {
17
18     public static void main(String[] args) throws SAXException, IOException, ParserConfigurati
19         // TODO Auto-generated method stub
20
21         //XML fájl meghívása
22         File xmlFile = new File("XMLW0E0RY.xml");
23
24         //Document builder definiálása
25         DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
26         DocumentBuilder dBuilder = factory.newDocumentBuilder();
27
28         //Fájl betöltése a documentum builderbe
29         Document doc = dBuilder.parse(xmlFile);
30         doc.getDocumentElement().normalize();
31
32         //Soforok kilistázása
33
34         System.out.println("");
35         System.out.println("Soforok kilistazasa: ");
36         System.out.println("");
37
38         NodeList soforList = doc.getElementsByTagName("Sofor");
39         for(int i=0;i<soforList.getLength();i++) {
40             Node nNode = soforList.item(i);
41             printSofor(nNode);
42         }
43
44
45
46         //Mercedes Tipusu autobuszok kilistazasa
47         System.out.println("");
48         System.out.println("Mercedes tipusu autobuszok: ");
49         System.out.println("");
50
51         NodeList feederList = doc.getElementsByTagName("Autobusz");
52         for(int i=0;i<feederList.getLength();i++) {
53             Node nNode = feederList.item(i);
54             printAutobusz(nNode, "Mercedes");
55         }
56
57
58         //Szervizek kilistázása másik megoldással
59
60         NodeList SzervizList =doc.getElementsByTagName("Szerviz");
61
62         for(int i=0;i<SzervizList.getLength();i++) {
63             Node nNode = SzervizList.item(i);
64             System.out.println("");
65             printSzerviz(nNode);
66         }
67
68
69
70

```

```

69
70
71 //kalauzok kilistázása
72
73 System.out.println("");
74 System.out.println("kalauzk kilistázása");
75
76 NodeList kalauzList = doc.getElementsByTagName("kalauz");
77 for(int i=0;i<kalauzList.getLength();i++) {
78     Node nNode = kalauzList.item(i);
79     System.out.println("");
80     printkalauz(nNode);
81 }
82
83 }
84
85
86
87
88 //Soforok kilistázása
89 private static void printSofor(Node nNode) {
90     if(nNode.getNodeType() == Node.ELEMENT_NODE) {
91         Element elem = (Element) nNode;
92         String SoforId = elem.getAttribute("SoforId");
93
94         Node nNode1 = elem.getElementsByTagName("Nev").item(0);
95         String Nev = nNode1.getTextContent();
96
97         System.out.printf("SoforID: %s\n", SoforId);
98         System.out.printf("Nev: %s\n", Nev);
99     }
100 }
101
102 //Mercedes típusu autobuszok kilistázása
103 private static void printAutobusz(Node nNode, String Tipus) {
104     if(nNode.getNodeType() == Node.ELEMENT_NODE) {
105         Element elem = (Element) nNode;
106         String AutobuszId = elem.getAttribute("AutobuszId");
107
108         Node nNode1 = elem.getElementsByTagName("Rendszam").item(0);
109         String Rendszam = nNode1.getTextContent();
110
111         Node nNode2 = elem.getElementsByTagName("Tipus").item(0);
112         String Tipus2 = nNode2.getTextContent();
113
114         Node nNode3 = elem.getElementsByTagName("max_hely").item(0);
115         String max_hely = nNode3.getTextContent();
116
117         if(Tipus.equals(Tipus2)) {
118             System.out.printf("AutobuszID: %s\n", AutobuszId);
119             System.out.printf("Rendszam: %s\n", Rendszam);
120             System.out.printf("Tipus %s\n", Tipus2);
121             System.out.printf("max_hely %s\n", max_hely);
122             System.out.println("");
123         }
124     }
125 }
126
127
128
129
130 //Szervizek kilistázása másik megoldással
131 private static void printSzerviz(Node nNode) {
132     if(nNode.getNodeType() == Node.ELEMENT_NODE) {
133         Element element = (Element) nNode;
134         NodeList nList = element.getChildNodes();
135         for (int j = 0; j < nList.getLength(); j++) {
136             Node node2 = nList.item(j);
137             if (node2.getNodeType() == Node.ELEMENT_NODE) {
138                 Element element2 = (Element) node2;
139                 System.out.println(node2.getNodeName()+" : "+node2.getTextContent());
140             }
141         }
142     }
143 }
144
145
146

```

```

146
147     //kalauzk kilistázása
148     private static void printkalauz(Node nNode) {
149         if(nNode.getNodeType()==Node.ELEMENT_NODE) {
150             Element element =(Element) nNode;
151             NodeList nList = element.getChildNodes();
152             for (int j = 0; j < nList.getLength(); j++) {
153                 Node node2 = nList.item(j);
154                 if (node2.getNodeType() == Node.ELEMENT_NODE) {
155                     Element element2 = (Element) node2;
156                     System.out.println(node2.getNodeName()+" : "+node2.getTextContent());
157
158
159                 }
160             }
161         }
162     }
163 }

```

Ezen programrész különböző lekérdezéseket futtat le.

Különböző adatokat dolgoz fel és értékeli a megadott kritériumok alapján.

## Az Adatmódosítás

Ebben a részben módosítottunk elemeket és azok attribútumait tulajdonságaikat, valamint a különböző tartalmakat.



```

1  package hudomparseW0E0RY;
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.OutputKeys;
10 import javax.xml.transform.Transformer;
11 import javax.xml.transform.TransformerConfigurationException;
12 import javax.xml.transform.TransformerException;
13 import javax.xml.transform.TransformerFactory;
14 import javax.xml.transform.dom.DOMSource;
15 import javax.xml.transform.stream.StreamResult;
16
17 import org.w3c.dom.Document;
18 import org.w3c.dom.Element;
19 import org.w3c.dom.Node;
20 import org.w3c.dom.NodeList;
21 import org.xml.sax.SAXException;
22
23 public class DomModifyW0E0RY {
24
25     public static void main(String[] args) throws SAXException, IOException, ParserConfigurationException {
26         // TODO Auto-generated method stub
27
28         //XML fájl meghívása
29         File xmlFile = new File("XMLW0E0RY.xml");
30
31         //Document builder definiálása
32         DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
33         DocumentBuilder dBuilder = factory.newDocumentBuilder();
34
35         //Fájl betöltése a DocumentBuilderbe
36         Document doc = dBuilder.parse(xmlFile);
37         doc.getDocumentElement().normalize();
38
39         //A 002-es id-val rendelkező Busz rendszamat ABC-123-ra változtatjuk
40         NodeList nodes = doc.getElementsByTagName("Busz");
41         for(int i=0;i<nodes.getLength();i++) {
42             Node node = nodes.item(i);
43             if(node.getNodeType() == Node.ELEMENT_NODE) {
44                 if(node.getAttributes().getNamedItem("BuszId").getTextContent().equals(
45                     NodeList childNodes = node.getChildNodes();
46                     for(int j=0;j<childNodes.getLength();j++) {
47                         Node childNode = childNodes.item(j);
48                         if(childNode.getNodeName().equals("Rendszam")) {
49                             childNode.setTextContent("ABC-123");
50                         }
51                     }
52                 }
53             }
54         }
55
56         //Újabb Buszt adunk hozzá
57         Element Busz = (Element)doc.getElementsByTagName("Busz").item(0);
58         Busz.appendChild(createBusz(doc,"010", "LKC-154", "Ikarus", "40"));
59
60         //Újabb Sofort adunk hozzá
61         Element Sofor = (Element)doc.getElementsByTagName("Soforok").item(0);
62         Sofor.appendChild(createSofor(doc,"L93751", "Korti Gyula", "086", "005"));
63
64         //Módosítjuk a 504-es id-val rendelkező kalauzt 601-re
65         modifyId(doc, "kalauz", "kalauzId", "504", " 601");
66
67         //Lementjük a módosított dokumentumot
68         SaveAsDoc(doc, "teszt.xml");
69     }
70
71     //Lementjük a módosított xml dokumentumot
72
73     public static void SaveAsDoc(Document doc, String filename) throws TransformerException {
74         TransformerFactory transformerFactory = TransformerFactory.newInstance();
75         Transformer transf = transformerFactory.newTransformer();
76

```

```

70     }
71
72     //Lementjük a módosított xml dokumentumot
73     public static void SaveAsDoc(Document doc, String filename) throws TransformerException {
74         TransformerFactory transformerFactory = TransformerFactory.newInstance();
75         Transformer transf = transformerFactory.newTransformer();
76
77         transf.setOutputProperty(OutputKeys.ENCODING, "UTF-8");
78         transf.setOutputProperty(OutputKeys.INDENT, "yes");
79         transf.setOutputProperty("{https://xml.apache.org/xslt}indent-amount", "2");
80
81         DOMSource source = new DOMSource(doc);
82
83         File myFile = new File(filename);
84
85         StreamResult console = new StreamResult(System.out);
86         StreamResult file = new StreamResult(myFile);
87
88         transf.transform(source, console);
89         transf.transform(source, file);
90     }
91
92     //Buszt hozunk létre
93     private static Node createBusz(Document doc, String BuszId, String Rendszam, String Tipus, String max_hely) {
94
95         Element user = doc.createElement("Busz");
96
97         user.setAttribute("BuszId", BuszId);
98         user.appendChild(createElement(doc, "Rendszam", Rendszam));
99         user.appendChild(createElement(doc, "Tipus", Tipus));
100        user.appendChild(createElement(doc, "max_hely", max_hely));
101
102        return user;
103    }
104
105
106    //Sofort hozunk létre
107    private static Node createSofor(Document doc, String SoforId, String Jegykiadoszam, String Nev, String Szektor) {
108
109        Element user = doc.createElement("Szektor");
110
111        user.setAttribute("SoforId", SoforId);
112        user.appendChild(createElement(doc, "Jegykiadoszam", Jegykiadoszam));
113        user.appendChild(createElement(doc, "Nev", Nev));
114        user.appendChild(createElement(doc, "BuszId_fk", BuszId_fk));
115
116
117        return user;
118    }
119
120    //Elementet hozunk létre
121    private static Node createElement(Document doc, String name, String value) {
122
123        Element node = doc.createElement(name);
124        node.appendChild(doc.createTextNode(value));
125
126        return node;
127    }
128
129    //Módosítjuk az id-t
130    public static void modifyId(Document doc, String tagName, String attrName, String oldData, String newData) {
131
132        NodeList nodes = doc.getElementsByTagName(tagName);
133        for (int i = 0; i < nodes.getLength(); i++) {
134            Node node = nodes.item(i);
135            if (node.getNodeType() == Node.ELEMENT_NODE) {
136                if (node.getAttribute(attrName).equals(oldData)) {
137                    node.setAttribute(attrName, newData);
138                }
139            }
140        }
141    }
142
143 }

```



## Az Adatírás

Itt láthatjuk azokat a program részleteket, amikkel kiíratunk konzolra különböző adatokat.

```
1  import org.w3c.dom.*;
2  import javax.xml.parsers.*;
3  import javax.xml.transform.*;
4  import javax.xml.transform.dom.DOMSource;
5  import javax.xml.transform.stream.StreamResult;
6  import java.io.File;
7
8  public class DOMWriteNeptunkod {
9      public static void main(String[] args) {
10         try {
11             // 1. DocumentBuilderFactory létrehozása
12             DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
13
14             // 2. DocumentBuilder létrehozása
15             DocumentBuilder builder = factory.newDocumentBuilder();
16
17             // 3. XML fájl betöltése
18             Document doc = builder.parse(new File("XMLNeptunkod.xml"));
19
20             // 4. Fastruktúra kiírása konzolra
21             printNode(doc.getDocumentElement(), 0);
22
23             // 5. Új XML fájl létrehozása és mentése
24             TransformerFactory transformerFactory = TransformerFactory.newInstance();
25             Transformer transformer = transformerFactory.newTransformer();
26             DOMSource source = new DOMSource(doc);
27             StreamResult result = new StreamResult(new File("XMLNeptunkod1.xml"));
28             transformer.transform(source, result);
29
30             System.out.println("XML fájl sikeresen létrehozva és elmentve: XMLNeptunkod1.xml");
31         } catch (Exception e) {
32             e.printStackTrace();
33         }
34     }
35
36     // Fastruktúra kiírása rekurzívan
37     private static void printNode(Node node, int indent) {
38         for (int i = 0; i < indent; i++) {
39             System.out.print(" ");
40         }
41         System.out.println(node.getNodeName() + ": " + node.getTextContent().trim());
42
43         NodeList nodeList = node.getChildNodes();
44         for (int i = 0; i < nodeList.getLength(); i++) {
45             Node childNode = nodeList.item(i);
46             if (childNode.getNodeType() == Node.ELEMENT_NODE) {
47                 printNode(childNode, indent + 1);
48             }
49         }
50     }
51 }
```