

# JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Telekommunikáció

Készítette: **Kató András**

Neptunkód: **S7KTW0**

Dátum: 2024.12.07.

# Tartalomjegyzék

Bevezetés.....	2
A feladat leírása.....	3
1. A telekommunikációs cég ábrázolása (ER, XDM, XML, XML Schema) .....	4
1.1 Az adatbázis ER modell tervezése .....	4
1.2 Az adatbázis konvertálása XDM modellre.....	5
1.3 Az XDM modell alapján XML dokumentum készítése .....	5
1.4 Az XML dokumentum alapján XML Schema készítése .....	9
2. DOM program készítése.....	13
2.1 adatolvasás .....	13
2.2 adatírás .....	18
2.3 adatlekérdezés .....	23
2.4 adاتمódosítás .....	26

## Bevezetés

Ezen jegyzőkönyv tartalmazza az Adatkezelés XML környezetben tárgy féléves beszámoló témájának a bemutatását, illetve megvalósítását. A dokumentum tartalmazni fogja a kódolást is, mely eltérő betűtípusban lesz elhelyezve, így könnyebbé téve az átláthatóságot. A dokumentum felépítése és stílusa megfelel a vele szemben elvárt formai követelményeknek.

## A feladat leírása

A féléves feladatom témájának a telekommunikációs cég bemutatását választottam. Mindig is érdekelt a téma és korábban dolgoztam egy ilyen cégnél, így szívesen készítenék, modelleznék egy ilyen adatbázist. A feladatomban 5 egyed szerepel: *Technológia; Szolgáltatás; Ügyfél; Dolgozó; Mentor*. Mindegyik egyednek van egy azonosítója (PK), több tulajdonsága és különböző kapcsolatban állnak egymással.

Az első egyed, melyet bemutatok az **Eszközgyártó** lesz. Ők gyártják a Szolgáltatások igénybevételéhez szükséges eszközöket, pl. routerek, STB-k. A tulajdonságai: gyarto\_id; nev; alapitasEve; bevetel.

A következő egyed a **Szolgáltatás** lesz, mely szoros kapcsolatban áll az Eszközgyártóval. Megtalálhatóak az igénybe vehető szolgáltatások nevei, árai és hogy milyen technológián vehetők igénybe. A tulajdonságai: szolg\_id ; nev; ar; technologia (pl. réz, koax, optika, mobil).

A harmadik egyed az **Ügyfél**. Az Ügyfélnek több tulajdonságát kell vizsgálnunk, pl. név, lakcím, szolgáltatások, életkor, szul\_ev – utóbbiak befolyással vannak arra, hogy mennyit fog fizetni az egyes szolgáltatásokért. A tulajdonságai: ugyfel\_id; nev; lakcim {irsz., kozseg, kozterulet, hazszam}; életkor; szul\_ev; aktivSzolgáltatások (pl. mobil, internet, TV) – többértékű tulajdonság.

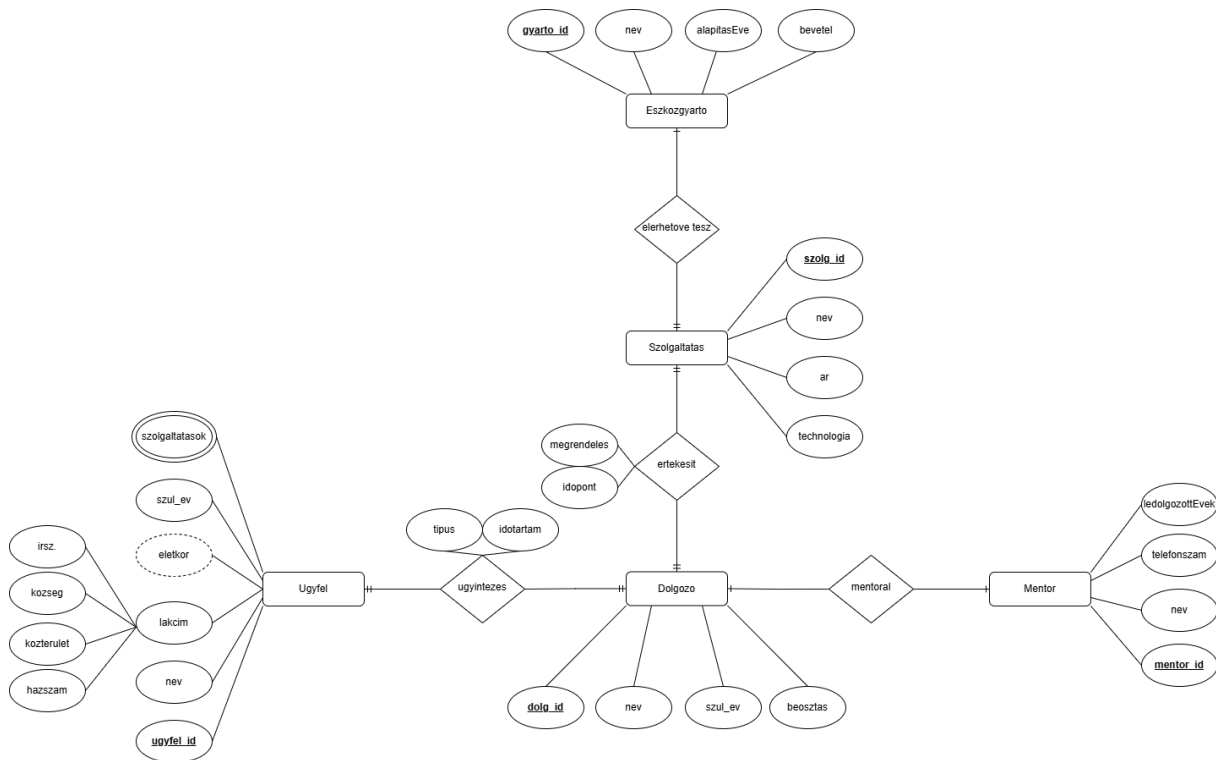
A **Dolgozó** egyed fogja tartalmazni a cégen belüli dolgozókat, a neveiket, születési évüket, beosztásukat. Egy Dolgozó és Ügyfél között több-több kapcsolat állhat fent (N:M). A tulajdonságai: dolgozo\_id; nev; szul\_ev; beosztas (pl. D2D, CC, üzlet, diakmunkas).

Az utolsó egyed a **Mentor**. Ők segítik a dolgozókat, növelik a cég hatékonyságát. Egy dolgozónak egy mentora van, és egy mentor csak egy dolgozót mentorálhat. A Mentor és Dolgozó kapcsolata remek példa lesz az 1:1 kapcsolatra. A tulajdonságai: mentor\_id; nev; telefonszam; ledolgozottEvek.

A felsorolt egyedeken felül, az N:M kapcsolatok esetén is vannak egyedi tulajdonságok, melyet az ER modell fog bemutatni a következő pontban!

# 1. A telekommunikációs cég ábrázolása (ER, XDM, XML, XML Schema)

## 1.1 Az adatbázis ER modell tervezése



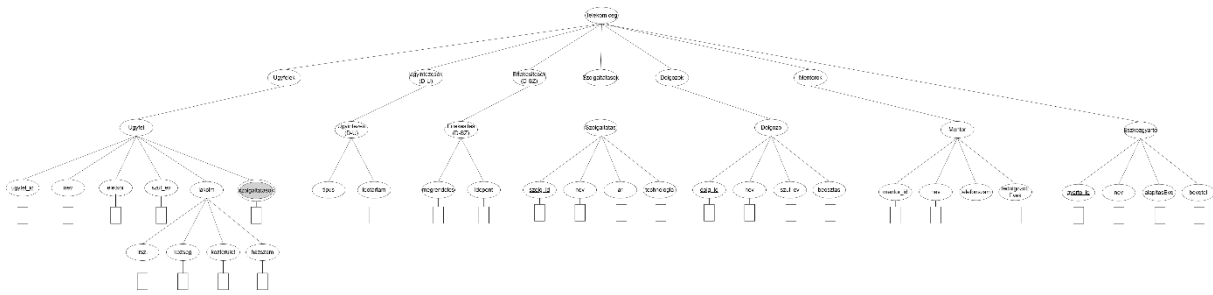
1. ábra: ER modell

Az ER modellemben megtalálható az 5 egyed. A Dolgozó és Mentor 1:1 kapcsolatban állnak, mivel egy mentor egy dolgozót mentorál és egy dolgozónak egy mentora van. Az 1:N kapcsolat is teljesül, egy Gyártó több Szolgáltatást is elérhetővé tesz az eszközeivel. Az M:N kapcsolat is teljesül, megfigyelhetjük az Ügyfél és Dolgozó között, vagy a Dolgozó és az értékesített Szolgáltatás között – egy dolgozó több szolgáltatást is értékesíthet és egy szolgáltatást több dolgozó árulhat.

Az kulcsokat aláhúzva jelöltem a modellben. Az Ügyfél egyed szolgáltatások tulajdonsága egy olyan tulajdonság, mely többértékű, hiszen lehet neki internet, TV, mobil szolgáltatása és ezek bármely kombinációja. Az Ügyfél életkora származtatott tulajdonság lesz, mivel kiszámolható, ha kivonjuk az éppen aktuális évből a születési évszámát.

Összegezvén az ER modell bemutatja, hogy az egyedek és a közöttük fellelhető kapcsolatok, valamint tulajdonságok hogyan kapcsolódnak és köthetőek egymáshoz.

## 1.2 Az adatbázis konvertálása XDM modellre



2. ábra: XDM modell

Az XDM modell az XML fájl alapja, így fontos volt az átláthatóság. Az 5 egyed ábrázolása mellett (*Eszközgyártó*, *Mentor*, *Dolgozó*, *Szolgáltatás*, *Ügyfél*) bővítettem a modellt két újabb egyeddel.

Ez a két új egyed a Dolgozó és Szolgáltatás közötti *Értékesítés*; valamint a Dolgozó és Ügyfél közötti *Ügyintézés*. Mivel ezen kapcsolatok tulajdonságokkal rendelkeznek, mindenképpen ábrázolni kellett őket a modellben. Az *Értékesítés* esetén (megrendelés, időpont), az *Ügyintézés* esetén (típus, időtartam).

Az így elkészül XDM modell alapján elkészíthető a telekommunikációs cég működését bemutató XML fájl.

## 1.3 Az XDM modell alapján XML dokumentum készítése

Az XML modell elkészítése egy nagyon fontos lépés, hiszen ez lapozza meg a további feladatok sikerességét.

Az elkészítését Eclipse IDE fejlesztő környezetben valósítottam meg. A vártnál tovább tartott, hogy felvázoljam az elképzelésemet és a struktúra kialakítása sem volt egyszerű, többször újra kellett kezdenem előlről, mivel amely úton haladtam, sajnos nem volt jó.

A következő beillesztett kódok bemutatják az XML kódomat, kommentekkel tűzdelve.

```
2 <?xml version="1.0" encoding="UTF-8"?>
3
4 <!-- Letrehozom a TelekomCeg-et -->
5 <TelekomCeg>
6
7     <!-- Definialom a dolgozokat, a tulajdonságait, attributumait -->
8     <Dolgozok>
9
10         <!-- Elso dolgozo -->
11         <Dolgozo dolg_id="D1">
12             <nev>Kerekes Andras</nev>
13             <szul_ev>1975</szul_ev>
14             <beosztas>D2D</beosztas>
15         </Dolgozo>
16
17         <!-- Masodik dolgozo -->
```

```

18         <beosztas>CC</beosztas>
19     <Dolgozo dolg_id="D2">
20         <nev>Juhász Ilona</nev>
21         <szul_ev>1985</szul_ev>
22     </Dolgozo>
23
24     <!-- Harmadik dolgozo -->
25     <Dolgozo dolg_id="D3">
26         <nev>Habar Aladar</nev>
27         <szul_ev>2000</szul_ev>
28         <beosztas>Gyakornok</beosztas>
29     </Dolgozo>
30
31 </Dolgozok>
32
33
34 <!-- Definialom a szolgáltatásokat, a tulajdonságait, attribútumait -->
35 <Szolgáltatások>
36
37     <!-- Elso szolgáltatás -->
38     <Szolgáltatás szolg_id="SZ1" d_sz="D1">
39         <nev>Internet</nev>
40         <ar>500</ar>
41         <technologia>optika</technologia>
42     </Szolgáltatás>
43
44     <!-- Masodik szolgáltatás -->
45     <Szolgáltatás szolg_id="SZ2" d_sz="D2">
46         <nev>TV</nev>
47         <ar>1500</ar>
48         <technologia>koax</technologia>
49     </Szolgáltatás>
50
51     <!-- Harmadik szolgáltatás -->
52     <Szolgáltatás szolg_id="SZ3" d_sz="D3">
53         <nev>Telefon</nev>
54         <ar>300</ar>
55         <technologia>rez</technologia>
56     </Szolgáltatás>
57
58     <!-- Negyedik szolgáltatás -->
59     <Szolgáltatás szolg_id="SZ4" d_sz="D2">
60         <nev>Mobil</nev>
61         <ar>5000</ar>
62         <technologia>radio</technologia>
63     </Szolgáltatás>
64
65 </Szolgáltatások>
66
67
68 <!-- Definialom az Ugyfeleket, a tulajdonságait, attribútumait -->
69 <Ugyfelek>
70
71     <!-- Elso ügyfél -->
72     <Ugyfél ugyfel_id="U1" d_u="D1">
73         <nev>Kovács Albert</nev>
74         <szul_ev>1990</szul_ev>
75         <eletkor>34</eletkor>
76         <lakcim>

```

```

77         <irsz>3500</irsz>
78         <kozseg>Miskolc</kozseg>
79         <kozterulet>Hejő utca</kozterulet>
80         <hazszam>10</hazszam>
81     </lakcim>
82     <aktivSzolgaltatasok>
83         <aktivSzolgaltatas>TV</aktivSzolgaltatas>
84         <aktivSzolgaltatas>Internet</aktivSzolgaltatas>
85     </aktivSzolgaltatasok>
86 </Ugyfel>
87
88 <!-- Masodik ugyfel -->
89 <Ugyfel ugyfel_id="U2" d_u="D2">
90     <nev>Nemeth Lili</nev>
91     <szul_ev>1970</szul_ev>
92     <eletkor>54</eletkor>
93     <lakcim>
94         <irsz>1200</irsz>
95         <kozseg>Budapest</kozseg>
96         <kozterulet>Rokon utca</kozterulet>
97         <hazszam>8</hazszam>
98     </lakcim>
99     <aktivSzolgaltatasok>
100         <aktivSzolgaltatas>Telefon</aktivSzolgaltatas>
101         <aktivSzolgaltatas>Mobil</aktivSzolgaltatas>
102     </aktivSzolgaltatasok>
103 </Ugyfel>
104
105 <!-- Harmadik ugyfel -->
106 <Ugyfel ugyfel_id="U3" d_u="D3">
107     <nev>Kato Andras</nev>
108     <szul_ev>2000</szul_ev>
109     <eletkor>24</eletkor>
110     <lakcim>
111         <irsz>3530</irsz>
112         <kozseg>Miskolc</kozseg>
113         <kozterulet>Sarolta utca</kozterulet>
114         <hazszam>8</hazszam>
115     </lakcim>
116     <aktivSzolgaltatasok>
117         <aktivSzolgaltatas>Internet</aktivSzolgaltatas>
118         <aktivSzolgaltatas>TV</aktivSzolgaltatas>
119         <aktivSzolgaltatas>Telefon</aktivSzolgaltatas>
120         <aktivSzolgaltatas>Mobil</aktivSzolgaltatas>
121     </aktivSzolgaltatasok>
122 </Ugyfel>
123
124 </Ugyfelek>
125
126
127 <!-- Definialom a mentorokat, a tulajdonsagait, attributumait -->
128 <Mentorok>
129
130     <!-- Elso mentor -->
131     <Mentor mentor_id="M1">
132         <nev>Hajdu Csilla</nev>
133         <telefonszam>06704682578</telefonszam>
134         <ledolgozottEvek>5</ledolgozottEvek>
135     </Mentor>

```

```

136
137     <!-- Masodik mentor -->
138     <Mentor mentor_id="M2">
139         <nev>Klinga Ferenc</nev>
140         <telefonszam>06304781236</telefonszam>
141         <ledolgozottEvek>8</ledolgozottEvek>
142     </Mentor>
143
144     <!-- Harmadik mentor -->
145     <Mentor mentor_id="M3">
146         <nev>Kecskes Milan</nev>
147         <telefonszam>06704286996</telefonszam>
148         <ledolgozottEvek>2</ledolgozottEvek>
149     </Mentor>
150
151 </Mentorok>
152
153
154 <!-- Definialom az eszkozgyartokat, a tulajdonsagait, attributumait -->
155 <Eszkozgyartok>
156
157     <!-- Elso eszkozgyarto -->
158     <Eszkozgyarto gyarto_id="ESZ1">
159         <nev>Bajor Tech.</nev>
160         <alapitasEve>1990</alapitasEve>
161         <bevetel>12000</bevetel>
162     </Eszkozgyarto>
163
164     <!-- Masodik eszkozgyarto -->
165     <Eszkozgyarto gyarto_id="ESZ2">
166         <nev>FinnTech</nev>
167         <alapitasEve>2010</alapitasEve>
168         <bevetel>150000</bevetel>
169     </Eszkozgyarto>
170
171     <!-- Harmadik eszkozgyarto -->
172     <Eszkozgyarto gyarto_id="ESZ3">
173         <nev>Clippy Technologies</nev>
174         <alapitasEve>1950</alapitasEve>
175         <bevetel>5000000</bevetel>
176     </Eszkozgyarto>
177
178 </Eszkozgyartok>
179
180
181 <!-- Definialom az ugyintezeseket, a tulajdonsagait, attributumait -->
182 <Ugyintezesek>
183
184     <!-- Elso ugyintezes -->
185     <Ugyintezes ugyintezes_id="UGY1">
186         <tipus>csomagvaltas</tipus>
187         <idotartam>12</idotartam>
188     </Ugyintezes>
189
190     <!-- Masodik ugyintezes -->
191     <Ugyintezes ugyintezes_id="UGY2">
192         <tipus>szerzodesKerdes</tipus>
193         <idotartam>5</idotartam>
194     </Ugyintezes>

```



```

195
196     <!-- Harmadik ugyintezes -->
197     <Ugyintezes ugyintezes_id="UGY3">
198         <tipus>lemondas</tipus>
199         <idotartam>30</idotartam>
200     </Ugyintezes>
201
202 </Ugyintezesek>
203
204
205 <!-- Definialom az értékesítéseket, a tulajdonságait, attribútumait -->
206 <Ertekesitesek>
207
208     <!-- Elso értékesites -->
209     <Ertekesites értékesites_id="ER1">
210         <megrendeles>TV</megrendeles>
211         <idopont>2024-12-07</idopont>
212     </Ertekesites>
213
214     <!-- Masodik értékesites -->
215     <Ertekesites értékesites_id="ER2">
216         <megrendeles>Mobil</megrendeles>
217         <idopont>2024-12-07</idopont>
218     </Ertekesites>
219
220     <!-- Harmadik értékesites -->
221     <Ertekesites értékesites_id="ER3">
222         <megrendeles>Internet</megrendeles>
223         <idopont>2024-12-07</idopont>
224     </Ertekesites>
225
226 </Ertekesitesek>
227
228 </TelekomCeg>

```

## 1.4 Az XML dokumentum alapján XML Schema készítése

Az XML Schema megvalósítása sem volt egyszerű, azonban az órákon tanultakat felhasználva sikerült megvalósítanom és validálnom is az XML fájlmot a soron következő Schemával.

A készítése során végig néztem a feladat követelményeit, és annak megfelelően bővítettem, illetve módosítottam. A jelenlegi Schema validálja az XML dokumentumot, valamint teljesít minden feladatrészt.

```

2  <?xml version="1.0" encoding="UTF-8"?>
3  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
4
5      <!-- A saját típusokat létrehozom, kesobb deifnialni fogom oket -->
6      <xs:element name="TelekomCeg">
7          <xs:complexType>
8              <xs:sequence>
9                  <xs:element name="Dolgozok" type="dolgozokTipus"/>
10                 <xs:element name="Szolgaltatasok" type="szolgaltatasokTipus"/>
11                 <xs:element name="Ugyfelek" type="ugyfelekTipus"/>
12                 <xs:element name="Mentorok" type="mentorokTipus"/>

```

```

13         <xs:element name="Eszkozgyartok" type="eszkozgyartokTipus"/>
14         <xs:element name="Ugyintezesek" type="ugyintezesekTipus"/>
15         <xs:element name="Ertekesitesek" type="ertekesitesekTipus"/>
16     </xs:sequence>
17 </xs:complexType>
18
19 <!-- Letrehozom a kulcsokat, key jelzo segitsegevel -->
20
21     <xs:key name="dolgozo_kulcs">
22         <xs:selector xpath="Dolgozo"/>
23         <xs:field xpath="@dolgozo_id"/>
24     </xs:key>
25
26     <xs:key name="szolgaltato_kulcs">
27         <xs:selector xpath="Szolgaltatas"/>
28         <xs:field xpath="@szolgaltato_id"/>
29     </xs:key>
30
31     <xs:key name="ugyfel_kulcs">
32         <xs:selector xpath="Ugyfel"/>
33         <xs:field xpath="@ugyfel_id"/>
34     </xs:key>
35
36     <xs:key name="mentor_kulcs">
37         <xs:selector xpath="Mentor"/>
38         <xs:field xpath="@mentor_id"/>
39     </xs:key>
40
41     <xs:key name="eszkozgyarto_kulcs">
42         <xs:selector xpath="Eszkozgyarto"/>
43         <xs:field xpath="@gyarto_id"/>
44     </xs:key>
45
46     <xs:key name="ugyintezes_kulcs">
47         <xs:selector xpath="Ugyintezes"/>
48         <xs:field xpath="@ugyintezes_id"/>
49     </xs:key>
50
51     <xs:key name="ertekesites_kulcs">
52         <xs:selector xpath="Ertekesites"/>
53         <xs:field xpath="@ertekesites_id"/>
54     </xs:key>
55
56     <!-- Referenciakat hozok lete. A dolgozo kapcsolatban van az
57     ugyfelekkel es szolgaltatasokkal -->
58
59     <xs:keyref name="szolg_dolgozo_ref" refer="dolgozo_kulcs">
60         <xs:selector xpath="Szolgaltatas"/>
61         <xs:field xpath="@dolgozo_id"/>
62     </xs:keyref>
63
64     <xs:keyref name="ugyfel_dolgozo_ref" refer="dolgozo_kulcs">
65         <xs:selector xpath="Ugyfel"/>
66         <xs:field xpath="@dolgozo_id"/>
67     </xs:keyref>
68
69     <!-- Egyedi, specialis definicio megvalositasa unique jelzovel
    dolgozo-ugyfelre -->

```

```

70         <xs:unique name="unique_ugyfel">
71             <xs:selector xpath="Ugyfel"/>
72             <xs:field xpath="@d_u"/>
73         </xs:unique>
74
75     </xs:element>
76
77     <!-- Dolgozok tipusának a definíálása, mivel több dolgozo van, ezért a
dolgozokTipusba illeszttem a dolgozoTipust -->
78     <xs:complexType name="dolgozokTipus">
79         <xs:sequence>
80             <xs:element name="Dolgozo" type="dolgozoTipus"
maxOccurs="unbounded"/>
81         </xs:sequence>
82     </xs:complexType>
83
84     <xs:complexType name="dolgozoTipus">
85         <xs:sequence>
86             <xs:element name="nev" type="xs:string"/>
87             <xs:element name="szul_ev" type="xs:integer"/>
88             <xs:element name="beosztas" type="xs:string"/>
89         </xs:sequence>
90         <xs:attribute name="dolg_id" type="xs:string" use="required"/>
91     </xs:complexType>
92
93     <!-- Szolgáltatások tipusának a definíálása -->
94     <xs:complexType name="szolgáltatásokTipus">
95         <xs:sequence>
96             <xs:element name="Szolgáltatás" type="szolgáltatásTipus"
maxOccurs="unbounded"/>
97         </xs:sequence>
98     </xs:complexType>
99
100    <xs:complexType name="szolgáltatásTipus">
101        <xs:sequence>
102            <xs:element name="nev" type="xs:string"/>
103            <xs:element name="ar" type="xs:integer"/>
104            <xs:element name="technologia" type="xs:string"/>
105        </xs:sequence>
106        <xs:attribute name="szolg_id" type="xs:string" use="required"/>
107        <xs:attribute name="d_sz" type="xs:string" use="required" />
108    </xs:complexType>
109
110    <!-- Ugyfelek tipusának a definíálása -->
111    <xs:complexType name="ugyfelekTipus">
112        <xs:sequence>
113            <xs:element name="Ugyfel" type="ugyfelTipus"
maxOccurs="unbounded"/>
114        </xs:sequence>
115    </xs:complexType>
116
117    <xs:complexType name="ugyfelTipus">
118        <xs:sequence>
119            <xs:element name="nev" type="xs:string"/>
120            <xs:element name="szul_ev" type="xs:integer"/>
121            <xs:element name="eletkor" type="xs:integer"/>
122            <xs:element name="lakcim" type="cimTipus"/>
123            <xs:element name="aktivSzolgáltatások" type="aktivSzolgTipus"/>
124        </xs:sequence>

```

```

125     <xs:attribute name="ugyfel_id" type="xs:string" use="required"/>
126     <xs:attribute name="d_u" type="xs:string" use="required" />
127 </xs:complexType>
128
129 <!-- Lakcim tipusának a definíálása, több elembol áll -->
130 <xs:complexType name="cimTipus">
131     <xs:sequence>
132         <xs:element name="irsz" type="xs:integer"/>
133         <xs:element name="kozseg" type="xs:string"/>
134         <xs:element name="kozterulet" type="xs:string"/>
135         <xs:element name="hazszam" type="xs:integer"/>
136     </xs:sequence>
137 </xs:complexType>
138
139 <!-- Aktiv szolgáltatások tipusának a definíálása, több elembol áll -->
140 <xs:complexType name="aktivSzolgTipus">
141     <xs:sequence>
142         <xs:element name="aktivSzolgaltatas" type="xs:string"
maxOccurs="unbounded"/>
143     </xs:sequence>
144 </xs:complexType>
145
146 <!-- Mentorok tipusának a definíálása -->
147 <xs:complexType name="mentorokTipus">
148     <xs:sequence>
149         <xs:element name="Mentor" type="mentorTipus"
maxOccurs="unbounded"/>
150     </xs:sequence>
151 </xs:complexType>
152
153 <xs:complexType name="mentorTipus">
154     <xs:sequence>
155         <xs:element name="nev" type="xs:string"/>
156         <xs:element name="telefonszam" type="xs:integer"/>
157         <xs:element name="ledolgozottEvek" type="xs:integer"/>
158     </xs:sequence>
159     <xs:attribute name="mentor_id" type="xs:string" use="required"/>
160 </xs:complexType>
161
162 <!-- Eszközigyártók tipusának a definíálása -->
163 <xs:complexType name="eszközigyartokTipus">
164     <xs:sequence>
165         <xs:element name="Eszközigyarto" type="eszközigyartoTipus"
maxOccurs="unbounded"/>
166     </xs:sequence>
167 </xs:complexType>
168
169 <xs:complexType name="eszközigyartoTipus">
170     <xs:sequence>
171         <xs:element name="nev" type="xs:string"/>
172         <xs:element name="alapitasEve" type="xs:integer"/>
173         <xs:element name="bevetel" type="xs:integer"/>
174     </xs:sequence>
175     <xs:attribute name="gyarto_id" type="xs:string" use="required"/>
176 </xs:complexType>
177
178 <!-- Ügyintézesek tipusának a definíálása -->
179 <xs:complexType name="ugyintezesekTipus">
180     <xs:sequence>

```

```

181         <xs:element name="Ugyintezes" type="ugyintezesTipus"
maxOccurs="unbounded"/>
182     </xs:sequence>
183 </xs:complexType>
184
185 <xs:complexType name="ugyintezesTipus">
186     <xs:sequence>
187         <xs:element name="tipus" type="xs:string"/>
188         <xs:element name="idotartam" type="xs:integer"/>
189     </xs:sequence>
190     <xs:attribute name="ugyintezes_id" type="xs:string" use="required"/>
191 </xs:complexType>
192
193 <!-- Ertekesitesek tipusának a definialasa -->
194 <xs:complexType name="ertekesitesekTipus">
195     <xs:sequence>
196         <xs:element name="Ertekesites" type="ertekesitesTipus"
maxOccurs="unbounded"/>
197     </xs:sequence>
198 </xs:complexType>
199
200 <xs:complexType name="ertekesitesTipus">
201     <xs:sequence>
202         <xs:element name="megrendeles" type="xs:string"/>
203         <xs:element name="idopont" type="xs:date"/>
204     </xs:sequence>
205     <xs:attribute name="ertekesites_id" type="xs:string" use="required"/>
206 </xs:complexType>
207
208 </xs:schema>

```

## 2. DOM program készítése

### 2.1 adatolvasás

Ezen részben a DOM Read elkészítésének a kódja látható. Az elkészítése nem volt egyszerű, azonban van egy kódrész, ami nagyon érdekes (135.sor). Ezen részben vizsgáljuk az ügyfél aktív szolgáltatásait. ELEMENT\_NODE esetén Element típusúvá alakítjuk, hogy később majd feltudjuk használni a tulajdonságot.

Utána megnézi a program az összes aktív szolgáltatást a második for ciklussal, és a getTextContent-el kinyeri a szöveges tartalmát, majd kiírja a konzolra és a fájlba is.

```

3 package hu.domparse.S7KTW0;
4
5 import java.io.File;
6 import java.io.IOException;
7 import java.io.FileWriter;
8 import java.io.BufferedWriter;
9 import javax.xml.parsers.DocumentBuilderFactory;
10 import javax.xml.parsers.DocumentBuilder;
11 import javax.xml.parsers.ParserConfigurationException;
12 import org.xml.sax.SAXException;
13 import org.w3c.dom.Document;
14 import org.w3c.dom.NodeList;
15 import org.w3c.dom.Node;

```

```

16 import org.w3c.dom.Element;
17
18 public class DOMReadS7KTW0 {
19
20     public static void main(String argv[]) throws SAXException, IOException,
        ParserConfigurationException {
21
22         // XMLS7KTW0.xml betoltes
23         File xmlFile = new File("XMLS7KTW0.xml");
24
25         // Letrehozom a dokumentum epitot es betoltom
26         DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
27         DocumentBuilder dBuilder = factory.newDocumentBuilder();
28
29         Document doc = dBuilder.parse(xmlFile);
30         doc.getDocumentElement().normalize();
31
32         // Ki kellesz irni fajlba, igy megadom mi legyen a neve a kiirt fajlnak
33         BufferedWriter writer = new BufferedWriter(new
        FileWriter("DOMReadS7KTW0out.txt"));
34
35         // Kiirom a gyoker elemet (TelekomCeg)
36         String rootElement = doc.getDocumentElement().getNodeName();
37         String output = "Gyoker elem: " + rootElement;
38
39         System.out.println(output);
40         writer.write(output);
41         writer.newLine();
42
43         // Kiiratom a dolgozokat
44         NodeList dolgozokList = doc.getElementsByTagName("Dolgozo");
45         for (int i = 0; i < dolgozokList.getLength(); i++) {
46
47             Node dolgozoNode = dolgozokList.item(i);
48
49             if (dolgozoNode.getNodeType() == Node.ELEMENT_NODE) {
50
51                 Element dolgozoElem = (Element) dolgozoNode;
52
53                 String dolg_id = dolgozoElem.getAttribute("dolg_id");
54                 String nev =
        dolgozoElem.getElementsByTagName("nev").item(0).getTextContent();
55                 String szul_ev =
        dolgozoElem.getElementsByTagName("szul_ev").item(0).getTextContent();
56                 String beosztas =
        dolgozoElem.getElementsByTagName("beosztas").item(0).getTextContent();
57
58                 output = "\nDolgozo ID: " + dolg_id + "\nNev: " + nev +
        "\nSzul_ev: " + szul_ev + "\nBeosztas: " + beosztas;
59
60                 System.out.println(output);
61                 writer.write(output);
62                 writer.newLine();
63             }
64         }
65
66         // Kiiratom a szolgáltatásokat
67         NodeList szolgáltatásokList = doc.getElementsByTagName("Szolgáltatás");
68         for (int i = 0; i < szolgáltatásokList.getLength(); i++) {

```

```

69
70         Node szolgáltatásNode = szolgáltatásokList.item(i);
71
72         if (szolgáltatásNode.getNodeType() == Node.ELEMENT_NODE) {
73
74             Element szolgáltatásElem = (Element) szolgáltatásNode;
75
76             String szolg_id = szolgáltatásElem.getAttribute("szolg_id");
77             String d_sz = szolgáltatásElem.getAttribute("d_sz");
78             String nev =
79 szolgáltatásElem.getElementsByTagName("nev").item(0).getTextContent();
80             String ar =
81 szolgáltatásElem.getElementsByTagName("ar").item(0).getTextContent();
82             String technologia =
83 szolgáltatásElem.getElementsByTagName("technologia").item(0).getTextContent();
84
85             output = "\nSzolgáltatás ID: " + szolg_id + "\nDolgozó ID: " +
86 d_sz + "\nNév: " + nev + "\nÁr: " + ar + "\nTechnológia: " + technologia;
87
88             System.out.println(output);
89             writer.write(output);
90             writer.newLine();
91         }
92     }
93
94     // Kiíratom az ügyfeleket
95     NodeList ügyfelekList = doc.getElementsByTagName("Ugyfel");
96     for (int i = 0; i < ügyfelekList.getLength(); i++) {
97
98         Node ügyfelNode = ügyfelekList.item(i);
99
100        if (ügyfelNode.getNodeType() == Node.ELEMENT_NODE) {
101
102            Element ügyfelElem = (Element) ügyfelNode;
103
104            String ügyfel_id = ügyfelElem.getAttribute("ügyfel_id");
105            String d_u = ügyfelElem.getAttribute("d_u");
106            String nev =
107 ügyfelElem.getElementsByTagName("nev").item(0).getTextContent();
108            String szul_ev =
109 ügyfelElem.getElementsByTagName("szul_ev").item(0).getTextContent();
110            String életkor =
111 ügyfelElem.getElementsByTagName("életkor").item(0).getTextContent();
112
113            output = "\nÜgyfel ID: " + ügyfel_id + "\nDolgozó ID: " + d_u +
114 "\nNév: " + nev + "\nSzül_ev: " + szul_ev + "\nÉletkor: " + életkor;
115
116            System.out.println(output);
117            writer.write(output);
118            writer.newLine();
119
120            // Lakcimek, mivel összetettek meg kell adni, vagy nem fog
121 megjeleni egyáltalán
122            Node lakcimNode = (Node)
123 ügyfelElem.getElementsByTagName("lakcim").item(0);
124
125            String irstz = ((Element)
126 lakcimNode).getElementsByTagName("irstz").item(0).getTextContent();

```

```

116         String kozseg = ((Element)
    lakcimNode).getElementsByTagName("kozseg").item(0).getTextContent();
117         String kozterulet = ((Element)
    lakcimNode).getElementsByTagName("kozterulet").item(0).getTextContent();
118         String hazszam = ((Element)
    lakcimNode).getElementsByTagName("hazszam").item(0).getTextContent();
119
120         output = "Lakcim: " + irsz + " " + kozseg + " " + kozterulet +
    " " + hazszam;
121
122         System.out.println(output);
123         writer.write(output);
124         writer.newLine();
125
126         // Mint a lakcimek, az aktiv szolgaltatasokat is kulon ki kell
irni
127         NodeList aktivSzolgaltatasokList =
    ugyfelElem.getElementsByTagName("aktivSzolgaltatasok");
128         for (int j = 0; j < aktivSzolgaltatasokList.getLength(); j++) {
129
130             Node aktivSzolgaltatasokNode =
    aktivSzolgaltatasokList.item(j);
131
132             if (aktivSzolgaltatasokNode.getNodeType() ==
    Node.ELEMENT_NODE) {
133                 Element aktivSzolgaltatas = (Element)
    aktivSzolgaltatasokNode;
134
135                 // Barmilyen variacio lehet, for-al megvizsgalom, es
kiiratom az osszes aktiv szolgaltatast
136                 NodeList szolgaltatasok =
    aktivSzolgaltatas.getElementsByTagName("aktivSzolgaltatas");
137                 for (int k = 0; k < szolgaltatasok.getLength(); k++) {
138
139                     Node szolgaltatasNode = szolgaltatasok.item(k);
140
141                     if (szolgaltatasNode.getNodeType() ==
    Node.ELEMENT_NODE) {
142                         String szolgaltatasNev =
    szolgaltatasNode.getTextContent();
143
144                         output = "Aktiv Szolgaltatas: " +
    szolgaltatasNev;
145
146                         System.out.println(output);
147                         writer.write(output);
148                         writer.newLine();
149                     }
150                 }
151             }
152         }
153     }
154 }
155
156 // Kiiratom a mentorokat
157 NodeList mentorokList = doc.getElementsByTagName("Mentor");
158
159 for (int i = 0; i < mentorokList.getLength(); i++) {
160

```



```

161         Node mentorNode = mentorokList.item(i);
162
163         if (mentorNode.getNodeType() == Node.ELEMENT_NODE) {
164
165             Element mentorElem = (Element) mentorNode;
166
167             String mentor_id = mentorElem.getAttribute("mentor_id");
168             String nev =
169                 mentorElem.getElementsByTagName("nev").item(0).getTextContent();
170             String telefonszam =
171                 mentorElem.getElementsByTagName("telefonszam").item(0).getTextContent();
172             String ledolgozottEvek =
173                 mentorElem.getElementsByTagName("ledolgozottEvek").item(0).getTextContent();
174
175             output = "\nMentor ID: " + mentor_id + "\nNev: " + nev +
176                 "\nTelefonszam: " + telefonszam + "\nLedolgozott Evek: " + ledolgozottEvek;
177
178             System.out.println(output);
179             writer.write(output);
180             writer.newLine();
181         }
182     }
183
184     // Kiíratom az eszkozgyartokat
185     NodeList eszkozgyartokList = doc.getElementsByTagName("Eszkozgyarto");
186     for (int i = 0; i < eszkozgyartokList.getLength(); i++) {
187
188         Node eszkozgyartoNode = eszkozgyartokList.item(i);
189
190         if (eszkozgyartoNode.getNodeType() == Node.ELEMENT_NODE) {
191
192             Element eszkozgyartoElem = (Element) eszkozgyartoNode;
193
194             String gyarto_id = eszkozgyartoElem.getAttribute("gyarto_id");
195             String nev =
196                 eszkozgyartoElem.getElementsByTagName("nev").item(0).getTextContent();
197             String alapitasEve =
198                 eszkozgyartoElem.getElementsByTagName("alapitasEve").item(0).getTextContent();
199             String bevetel =
200                 eszkozgyartoElem.getElementsByTagName("bevetel").item(0).getTextContent();
201
202             output = "\nEszkozgyarto ID: " + gyarto_id + "\nNev: " + nev +
203                 "\nAlapitas eve: " + alapitasEve + "\nBevetel: " + bevetel;
204
205             System.out.println(output);
206             writer.write(output);
207             writer.newLine();
208         }
209     }
210
211     // Kiíratom az ugyintezeseket
212     NodeList ugyintezesekList = doc.getElementsByTagName("Ugyintezes");
213     for (int i = 0; i < ugyintezesekList.getLength(); i++) {
214
215         Node ugyintezesNode = ugyintezesekList.item(i);
216
217         if (ugyintezesNode.getNodeType() == Node.ELEMENT_NODE) {
218
219             Element ugyintezesElem = (Element) ugyintezesNode;

```

```

212
213         String ugyintezes_id =
    ugyintezesElem.getAttribute("ugyintezes_id");
214         String tipus =
    ugyintezesElem.getElementsByTagName("tipus").item(0).getTextContent();
215         String idotartam =
    ugyintezesElem.getElementsByTagName("idotartam").item(0).getTextContent();
216
217         output = "\nUgyintezes ID: " + ugyintezes_id + "\nTipus: " +
    tipus + "\nIdotartam: " + idotartam;
218
219         System.out.println(output);
220         writer.write(output);
221         writer.newLine();
222     }
223 }
224
225 // Kiíratom az értékesítéseket
226 NodeList ertekesitesekList = doc.getElementsByTagName("Ertekesites");
227 for (int i = 0; i < ertekesitesekList.getLength(); i++) {
228
229     Node ertekesitesNode = ertekesitesekList.item(i);
230
231     if (ertekesitesNode.getNodeType() == Node.ELEMENT_NODE) {
232
233         Element ertekesitesElem = (Element) ertekesitesekList.item(i);
234
235         String ertekesites_id =
    ertekesitesElem.getAttribute("ertekesites_id");
236         String megrendeles =
    ertekesitesElem.getElementsByTagName("megrendeles").item(0).getTextContent();
237         String idopont =
    ertekesitesElem.getElementsByTagName("idopont").item(0).getTextContent();
238
239         output = "\nErtekesites ID: " + ertekesites_id +
    "\nMegrendeles: " + megrendeles + "\nIdopont: " + idopont;
240
241         System.out.println(output);
242         writer.write(output);
243         writer.newLine();
244     }
245 }
246
247 // fajliras vege
248 writer.close();
249 }
250 }

```

## 2.2 adatírás

Ezen DOM Write program faszervezet formában írja ki az egyedeket, valamint azok attribútumait és tulajdonságait. Ami számomra érdekes ebben a kiírásban, hogy amilyen rendezett a konzolon a tartalom, maga a kód is ugyanolyan áttekinthető.

Például az Eszközgyártók kiiratásánál (156), látható, hogy milyen a felépítése ennek a programnak. Maga a program fájlba is kiírja az adatokat, melyet XMLS7KTW01.xml néven lehet megtalálni.

```
3 package hu.domparse.S7KTW0;
4
5 import java.io.File;
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.TransformerException;
12 import javax.xml.transform.TransformerFactory;
13 import javax.xml.transform.dom.DOMSource;
14 import javax.xml.transform.stream.StreamResult;
15 import org.w3c.dom.Document;
16 import org.w3c.dom.Element;
17 import org.w3c.dom.Node;
18
19 public class DOMWriteS7KTW0 {
20
21     public static void main(String[] args) throws ParserConfigurationException,
22     TransformerException {
23
24         // Letrehozom a dokumentum építőt
25         DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
26         DocumentBuilder builder = factory.newDocumentBuilder();
27         Document doc = builder.newDocument();
28
29         // Letrehozom a gyökerelemet
30         Element root = doc.createElementNS("DOMS7KTW0", "TelekomCeg");
31         doc.appendChild(root);
32
33         // Hozzaadom az egyedeket
34         root.appendChild(createDoLgozok(doc));
35         root.appendChild(createSzolGaltatasok(doc));
36         root.appendChild(createUgyfeLek(doc));
37         root.appendChild(createMentorok(doc));
38         root.appendChild(createEszkozgyartok(doc));
39         root.appendChild(createUgyintezesek(doc));
40         root.appendChild(createErtekesitesek(doc));
41
42         // Ki kell a feladatnak megfeleloen irni XML fajlkent
43         TransformerFactory transformerFactory =
44         TransformerFactory.newInstance();
45         Transformer transf = transformerFactory.newTransformer();
46
47         transf.setOutputProperty(OutputKeys.ENCODING, "UTF-8");
48         transf.setOutputProperty(OutputKeys.INDENT, "yes");
49
50         DOMSource source = new DOMSource(doc);
51         File myFile = new File("XMLS7KTW01.xml");
52
53         StreamResult console = new StreamResult(System.out);
54         StreamResult file = new StreamResult(myFile);
55
56         //konzolra is es fajlba is
57         transf.transform(source, console);
```

```

56         transf.transform(source, file);
57     }
58
59     //Kiiratom a dolgozokat
60     private static Node createDolgozok(Document doc) {
61         Element dolgozok = doc.createElement("Dolgozok");
62
63         dolgozok.appendChild(createDolgozo(doc, "D1", "Kerekes Andras", "1975",
64 "D2D"));
65         dolgozok.appendChild(createDolgozo(doc, "D2", "Juhász Ilona", "1985",
66 "CC"));
67         dolgozok.appendChild(createDolgozo(doc, "D3", "Habar Aladar", "2000",
68 "Gyakornok"));
69         return dolgozok;
70     }
71     private static Node createDolgozo(Document doc, String id, String nev,
72 String szulEv, String beosztas) {
73         Element dolgozo = doc.createElement("Dolgozo");
74         dolgozo.setAttribute("dolg_id", id);
75
76         dolgozo.appendChild(createElement(doc, "nev", nev));
77         dolgozo.appendChild(createElement(doc, "szul_ev", szulEv));
78         dolgozo.appendChild(createElement(doc, "beosztas", beosztas));
79         return dolgozo;
80     }
81
82     // Kiiratom a szolgáltatásokat
83     private static Node createSzolgáltatások(Document doc) {
84         Element szolgáltatások = doc.createElement("Szolgáltatások");
85
86         szolgáltatások.appendChild(createSzolgáltatás(doc, "SZ1", "D1",
87 "Internet", "500", "optika"));
88         szolgáltatások.appendChild(createSzolgáltatás(doc, "SZ2", "D2", "TV",
89 "1500", "koax"));
90         szolgáltatások.appendChild(createSzolgáltatás(doc, "SZ3", "D3",
91 "Telefon", "300", "rez"));
92         szolgáltatások.appendChild(createSzolgáltatás(doc, "SZ4", "D2",
93 "Mobil", "5000", "radio"));
94         return szolgáltatások;
95     }
96     private static Node createSzolgáltatás(Document doc, String id, String
97 d_sz, String nev, String ar, String technologia) {
98         Element szolgáltatás = doc.createElement("Szolgáltatás");
99         szolgáltatás.setAttribute("szolg_id", id);
100         szolgáltatás.setAttribute("d_sz", d_sz);
101
102         szolgáltatás.appendChild(createElement(doc, "nev", nev));
103         szolgáltatás.appendChild(createElement(doc, "ar", ar));
104         szolgáltatás.appendChild(createElement(doc, "technologia",
105 technologia));
106         return szolgáltatás;
107     }
108
109     // Kiiratom az ügyfeleket
110     private static Node createUgyfelek(Document doc) {
111         Element ugyfelek = doc.createElement("Ugyfelek");
112     }

```

```

105     ugyfelek.appendChild(createUgyfel(doc, "U1", "D1", "Kovacs Albert",
    "1990", "34", "3500", "Miskolc", "Hejo utca", "10", new String[]{"TV",
    "Internet"}));
106     ugyfelek.appendChild(createUgyfel(doc, "U2", "D2", "Nemeth Lili",
    "1970", "54", "1200", "Budapest", "Rokon utca", "8", new String[]{"Telefon",
    "Mobil"}));
107     ugyfelek.appendChild(createUgyfel(doc, "U3", "D3", "Kato Andras",
    "2000", "24", "3530", "Miskolc", "Sarolta utca", "8", new String[]{"Internet",
    "TV", "Telefon", "Mobil"}));
108     return ugyfelek;
109 }
110 private static Node createUgyfel(Document doc, String id, String d_u,
    String nev, String szulEv, String eletkor,
111     String irsz, String kozseg, String
    kozterulet, String hazszam, String[] aktivSzolgaltatasok) {
112     Element ugyfel = doc.createElement("Ugyfel");
113     ugyfel.setAttribute("ugyfel_id", id);
114     ugyfel.setAttribute("d_u", d_u);
115
116     ugyfel.appendChild(createElement(doc, "nev", nev));
117     ugyfel.appendChild(createElement(doc, "szul_ev", szulEv));
118     ugyfel.appendChild(createElement(doc, "eletkor", eletkor));
119
120     Element lakcim = doc.createElement("lakcim");
121     lakcim.appendChild(createElement(doc, "irsz", irsz));
122     lakcim.appendChild(createElement(doc, "kozseg", kozseg));
123     lakcim.appendChild(createElement(doc, "kozterulet", kozterulet));
124     lakcim.appendChild(createElement(doc, "hazszam", hazszam));
125     ugyfel.appendChild(lakcim);
126
127     Element aktivSzolg = doc.createElement("aktivSzolgaltatasok");
128     for (String szolg : aktivSzolgaltatasok) {
129         aktivSzolg.appendChild(createElement(doc, "aktivSzolgaltatas",
    szolg));
130     }
131     ugyfel.appendChild(aktivSzolg);
132     return ugyfel;
133 }
134
135
136 // Kiiratom a mentorokat
137 private static Node createMentorok(Document doc) {
138     Element mentorok = doc.createElement("Mentorok");
139
140     mentorok.appendChild(createMentor(doc, "M1", "Hajdu Csilla",
    "06704682578", "5"));
141     mentorok.appendChild(createMentor(doc, "M2", "Klinga Ferenc",
    "06304781236", "8"));
142     mentorok.appendChild(createMentor(doc, "M3", "Kecskes Milan",
    "06704286996", "2"));
143     return mentorok;
144 }
145 private static Node createMentor(Document doc, String id, String nev,
    String telefonszam, String ledolgozottEvek) {
146     Element mentor = doc.createElement("Mentor");
147     mentor.setAttribute("mentor_id", id);
148
149     mentor.appendChild(createElement(doc, "nev", nev));
150     mentor.appendChild(createElement(doc, "telefonszam", telefonszam));

```

```

151         mentor.appendChild(createElement(doc, "ledolgozottEvek",
        ledolgozottEvek));
152         return mentor;
153     }
154
155
156     // Kiiratom az eszkozgyartokat
157     private static Node createEszkozgyartok(Document doc) {
158         Element eszkozgyartok = doc.createElement("Eszkozgyartok");
159
160         eszkozgyartok.appendChild(createEszkozgyarto(doc, "ESZ1", "Bajor
        Tech.", "1990", "12000"));
161         eszkozgyartok.appendChild(createEszkozgyarto(doc, "ESZ2", "FinnTech",
        "2010", "150000"));
162         eszkozgyartok.appendChild(createEszkozgyarto(doc, "ESZ3", "Clippy
        Technologies", "1950", "5000000"));
163         return eszkozgyartok;
164     }
165     private static Node createEszkozgyarto(Document doc, String id, String nev,
        String alapitasEve, String bevetel) {
166         Element eszkozgyarto = doc.createElement("Eszkozgyarto");
167         eszkozgyarto.setAttribute("gyarto_id", id);
168
169         eszkozgyarto.appendChild(createElement(doc, "nev", nev));
170         eszkozgyarto.appendChild(createElement(doc, "alapitasEve",
        alapitasEve));
171         eszkozgyarto.appendChild(createElement(doc, "bevetel", bevetel));
172         return eszkozgyarto;
173     }
174
175
176     // Kiiratom az ugyintezeseket
177     private static Node createUgyintezesek(Document doc) {
178         Element ugyintezesek = doc.createElement("Ugyintezesek");
179
180         ugyintezesek.appendChild(createUgyintezes(doc, "UGY1", "csomagvaltas",
        "12"));
181         ugyintezesek.appendChild(createUgyintezes(doc, "UGY2",
        "szerzodesKerdes", "5"));
182         ugyintezesek.appendChild(createUgyintezes(doc, "UGY3", "lemondas",
        "30"));
183         return ugyintezesek;
184     }
185     private static Node createUgyintezes(Document doc, String id, String tipus,
        String idotartam) {
186         Element ugyintezes = doc.createElement("Ugyintezes");
187         ugyintezes.setAttribute("ugyintezes_id", id);
188
189         ugyintezes.appendChild(createElement(doc, "tipus", tipus));
190         ugyintezes.appendChild(createElement(doc, "idotartam", idotartam));
191         return ugyintezes;
192     }
193
194
195     // Kiiratom az ertekesiteseket
196     private static Node createErtekesitesek(Document doc) {
197         Element ertekesitesek = doc.createElement("Ertekesitesek");
198

```

```

199     ertekezesek.appendChild(createErtekesites(doc, "ER1", "TV", "2024-12-
    07"));
200     ertekezesek.appendChild(createErtekesites(doc, "ER2", "Mobil", "2024-
    12-04"));
201     ertekezesek.appendChild(createErtekesites(doc, "ER3", "Internet",
    "2024-12-05"));
202     return ertekezesek;
203 }
204 private static Node createErtekesites(Document doc, String id, String
    megrendeles, String idopont) {
205     Element ertekesites = doc.createElement("Ertekesites");
206     ertekesites.setAttribute("ertekesites_id", id);
207
208     ertekesites.appendChild(createElement(doc, "megrendeles",
    megrendeles));
209     ertekesites.appendChild(createElement(doc, "idopont", idopont));
210     return ertekesites;
211 }
212
213 //letrehozzuk az XML-t es visszaadjuk returnel
214 private static Node createElement(Document doc, String name, String value)
    {
215     Element node = doc.createElement(name);
216     node.appendChild(doc.createTextNode(value));
217     return node;
218 }
219 }

```

## 2.3 adatlekérdezés

Az adat lekérdező DOM Query program megvalósítja a 4 lekérdezést. Ezek különböznek egymástól. Hasonló lekérdezéseket ismerhettünk meg a félév során, az órarendek, illetve hallgatók, kurzusfelvétellel kapcsolatos XML fájlkon keresztül.

Az összetett lekérdezés (109) ismerős lehet a gyakorlati órákon tanultakból. Egy összesített lista látható, mely kiválasztja az eszközgyártók és mentorok nevét, for ciklussal végig megy rajtuk és kiolvassa őket.

```

3 package hu.domparse.S7KTW0;
4
5
6 import java.io.File;
7
8 import javax.xml.parsers.DocumentBuilder;
9 import javax.xml.parsers.DocumentBuilderFactory;
10
11 import org.w3c.dom.Document;
12 import org.w3c.dom.Element;
13 import org.w3c.dom.Node;
14 import org.w3c.dom.NodeList;
15
16 public class DOMQueryS7KTW0 {
17
18     public static void main(String[] args) {
19
20         try {

```



```

21
22      // XMLS7KTW0.xml betoltes
23      File inputFile = new File("XMLS7KTW0.xml");
24
25      // Letrehozom a dokumentum epitot es betoltom
26      DocumentBuilderFactory dbFactory =
DocumentBuilderFactory.newInstance();
27      DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
28      Document doc = dBuilder.parse(inputFile);
29
30      doc.getDocumentElement().normalize();
31
32
33
34      // 1. Kiiratom a dolgozok nevet
35
36      System.out.println("1. A dolgozok nevei: ");
37      NodeList dolgozok = doc.getElementsByTagName("Dolgozo");
38
39      for (int i = 0; i < dolgozok.getLength(); i++) {
40
41          Node node = dolgozok.item(i);
42
43          if (node.getNodeType() == Node.ELEMENT_NODE) {
44
45              Element nevek = (Element) node;
46
47              System.out.println(nevek.getElementsByTagName("nev").item(0).getTextContent());
48          }
49      }
50
51
52      // 2. Kiiratom a szolgaltatasok nevet es arat
53
54      System.out.println("\n2. A szolgaltatsok neve es az araik: ");
55      NodeList szolgaltatasok = doc.getElementsByTagName("Szolgaltatas");
56
57      for (int i = 0; i < szolgaltatasok.getLength(); i++) {
58
59          Node node = szolgaltatasok.item(i);
60
61          if (node.getNodeType() == Node.ELEMENT_NODE) {
62
63              Element NevAR = (Element) node;
64              String nev =
NevAR.getElementsByTagName("nev").item(0).getTextContent();
65              String ar =
NevAR.getElementsByTagName("ar").item(0).getTextContent();
66              System.out.println("Nev: " + nev + ", Ar: " + ar);
67          }
68      }
69
70
71
72      // 3. Kiiratom a harmadik ugyfel adatait
73
74      System.out.println("\n3. A harmadik ugyfel adatai: ");
75      NodeList ugyfelek = doc.getElementsByTagName("Ugyfel");

```



```

76
77         for (int i = 0; i < ugyfelek.getLength(); i++) {
78
79             Node node = ugyfelek.item(i);
80
81             if (node.getNodeType() == Node.ELEMENT_NODE) {
82
83                 Element harmadikUgyfel = (Element) node;
84
85                 if
86 (harmadikUgyfel.getElementsByTagName("nev").item(0).getTextContent().equals("Ka
to Andras")) {
87
88                     System.out.println("Nev: " +
harmadikUgyfel.getElementsByTagName("nev").item(0).getTextContent());
89                     System.out.println("Szuletesi ev: " +
harmadikUgyfel.getElementsByTagName("szul_ev").item(0).getTextContent());
90                     System.out.println("Eletkor: " +
harmadikUgyfel.getElementsByTagName("eletkor").item(0).getTextContent());
91                     Element lakcim = (Element)
harmadikUgyfel.getElementsByTagName("lakcim").item(0);
92                     System.out.println("Lakcime: " +
lakcim.getElementsByTagName("irsz").item(0).getTextContent() + ", " +
93 lakcim.getElementsByTagName("kozseg").item(0).getTextContent() + ", " +
94 lakcim.getElementsByTagName("kozterulet").item(0).getTextContent() + " " +
95 lakcim.getElementsByTagName("hazszam").item(0).getTextContent());
96
97                     NodeList aktivSzolgaltatasok =
harmadikUgyfel.getElementsByTagName("aktivSzolgaltatas");
98                     System.out.println("Aktiv szolgaltatasok: ");
99
100                     for (int j = 0; j < aktivSzolgaltatasok.getLength();
j++) {
101
102                         System.out.println(" - " +
aktivSzolgaltatasok.item(j).getTextContent());
103                     }
104                 }
105             }
106
107
108
109         // 4. Összetett lekerdezes: mentorok es eszkozgyartok nevei
110
111         System.out.println("\n4. Mentorok es eszkozgyartok nevei:");
112         NodeList mentorok = doc.getElementsByTagName("Mentor");
113         NodeList eszkozgyartok = doc.getElementsByTagName("Eszkozgyarto");
114
115         for (int i = 0; i < Math.max(mentorok.getLength(),
eszkozgyartok.getLength()); i++) {
116
117             String mentorNev = i < mentorok.getLength() ? ((Element)
mentorok.item(i)).getElementsByTagName("nev").item(0).getTextContent() : "-";

```

```

118         String gyartoNev = i < eszkozgyartok.getLength() ? ((Element)
    eszkozgyartok.item(i)).getElementsByTagName("nev").item(0).getTextContent() :
    "-";
119         System.out.println("Mentor: " + mentorNev + ", Eszkozgyarto: "
    + gyartoNev);
120     }
121
122     }
123     catch (Exception e) {
124
125         e.printStackTrace();
126     }
127 }
128 }

```

## 2.4 adatmódosítás

Az utolsó feladatrész, mely a DOM Modify, vagyis az adatmódosítás. Ezen részben találhatóak olyan módosítások, melyeket szintén láthattunk a félév során. Egy különleges része a feladatnak, a perc kiírása a Ügyintézési időtartamnak (73).

Ez a rész azért érdekes, mert az eredeti XML fájlban a típus stringnek lett hagyva az időtartamnak. Az XML fájlban csak számot adtam meg, mivel tervem volt vele, hogy azt majd később módosítom és időegységet írok mögé. Pontosan ezt csinálja a hivatkozott kódrész, a „perc” stringet illeszti be a számok mögé.

```

3  package hu.domparse.S7KTW0;
4
5  import java.io.File;
6  import javax.xml.parsers.DocumentBuilder;
7  import javax.xml.parsers.DocumentBuilderFactory;
8  import javax.xml.transform.Transformer;
9  import javax.xml.transform.TransformerFactory;
10 import javax.xml.transform.dom.DOMSource;
11 import javax.xml.transform.stream.StreamResult;
12
13 import org.w3c.dom.Document;
14 import org.w3c.dom.Element;
15 import org.w3c.dom.Node;
16 import org.w3c.dom.NodeList;
17
18 public class DOMModifyS7KTW0 {
19
20     public static void main(String[] args) {
21
22         try {
23
24             // XMLS7KTW0.xml betoltes
25             File inputFile = new File("XMLS7KTW0.xml");
26
27             // Letrehozom a dokumentum epitot es betoltom
28             DocumentBuilderFactory docFactory =
    DocumentBuilderFactory.newInstance();
29             DocumentBuilder docBuilder = docFactory.newDocumentBuilder();
30             Document doc = docBuilder.parse(inputFile);
31             doc.getDocumentElement().normalize();

```

```

32
33
34
35 // 1. Az első es második dolgozó nevének a csereje
36 NodeList dolgozok = doc.getElementsByTagName("Dolgozo");
37 Node elsőDolgozo = dolgozok.item(0);
38 Node harmadikDolgozo = dolgozok.item(2);
39
40 if (elsőDolgozo.getNodeType() == Node.ELEMENT_NODE &&
harmadikDolgozo.getNodeType() == Node.ELEMENT_NODE) {
41
42     Element elsőDolgozoD1 = (Element) elsőDolgozo;
43     Element harmadikDolgozoD3 = (Element) harmadikDolgozo;
44
45     String elsőNév =
elsőDolgozoD1.getElementsByTagName("nev").item(0).getTextContent();
46     String harmadikNév =
harmadikDolgozoD3.getElementsByTagName("nev").item(0).getTextContent();
47
48     elsőDolgozoD1.getElementsByTagName("nev").item(0).setTextContent(harmadikNév);
49     harmadikDolgozoD3.getElementsByTagName("nev").item(0).setTextContent(elsőNév);
50 }
51
52
53
54 // 2. Az első szolgáltató nevének az átírása
55 NodeList szolgáltatasok = doc.getElementsByTagName("Szolgáltatas");
56 Node elsőSzolgáltatas = szolgáltatasok.item(0);
57
58 if (elsőSzolgáltatas.getNodeType() == Node.ELEMENT_NODE) {
59
60     Element elsőSzolgáltatasSZ1 = (Element) elsőSzolgáltatas;
61     elsőSzolgáltatasSZ1.getElementsByTagName("nev").item(0).setTextContent("DSL");
62 }
63
64
65
66 // 3. Az első ügyfél adatainak a törlese
67 NodeList ügyfelek = doc.getElementsByTagName("Ugyfel");
68 Node elsőUgyfel = ügyfelek.item(0);
69 elsőUgyfel.getParentNode().removeChild(elsőUgyfel);
70
71
72
73 // 4. Ügyintézési időtartam számok mögé szöveg kiírása (perc)
74 NodeList ügyintezesek = doc.getElementsByTagName("Ugyintezes");
75 for (int i = 0; i < ügyintezesek.getLength(); i++) {
76
77     Node ügyintezes = ügyintezesek.item(i);
78
79     if (ügyintezes.getNodeType() == Node.ELEMENT_NODE) {
80
81         Element ügyintezesPerc = (Element) ügyintezes;
82         String időtartam =
ügyintezesPerc.getElementsByTagName("idotartam").item(0).getTextContent();

```

```

83      ugyintezesPerc.getElementsByTagName("idotartam").item(0).setTextContent(idotart
am + " perc");
84      }
85      }
86
87      // Mentes es konzolra iras
88      TransformerFactory transformerFactory =
TransformerFactory.newInstance();
89      Transformer transformer = transformerFactory.newTransformer();
90      DOMSource source = new DOMSource(doc);
91
92      StreamResult consoleResult = new StreamResult(System.out);
93      transformer.transform(source, consoleResult);
94
95      }
96      catch (Exception e) {
97
98          e.printStackTrace();
99      }
100  }
101 }

```