

# JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Logisztikai Cég

Készítette: **Nagy Balázs**  
Neptunkód: **EIO1RQ**  
Dátum: **2023.12.05**

# Tartalomjegyzék

Bevezetés .....	3
1. Feladat .....	4
a) Az adatbázis ER modell tervezése .....	4
b) Az adatbázis konvertálása XDM modellre .....	6
c) Az XDM modell alapján XML dokumentum készítése .....	7
d) Az XML dokumentum alapján XMLSchema készítése- saját típusok, ref, key, keyref, speciális elemek .....	13
2. Feladat .....	19
2a) Adatolvasás .....	19
2b) Adatmódosítás (kód - comment együtt) fájlnev: DOMModifyNeptunkod.java .....	21
2c) Adatlekérdezés .....	23
2d) Adatírás -készítsen egy DOM API programot, amely egy XMLNeptunkod.xml dokumentum tartalmát fa struktúra formában kiírja a konzolra és egy XMLNeptunkod1.xml fájlba.(kód –comment együtt)–fájlnev: DOMWriteNeptunkod.java .....	26

## Bevezetés

**A feladat leírása:** Egy logisztikai cég raktárépületéhez/telephelyéhez készült adatbázis, amely reprezentálja az említett raktár hierarchikus felépítését a dolgozók és a részlegek között.

A részlegek itt különböző típusú árucikkekkel dolgoznak (pl.: Italok, étel, dohány, stb.) és csupán azért felelnek, hogy a beérkező megrendelések alapján összekészítsék a termékeket szállításra; az árú szállítását már nem a példában szereplő cég végzi, Így az adatbázis nem tartalmaz a szállítással kapcsolatos információkat a szállítási helyet leszámítva, amit a cég továbbít a szállításért felelős szervezetnek. Az adatbázis részletes információkat tartalmaz az egyes megrendelésekről és azok vevőiről is egyaránt.

Maga az adatbázis elősegíti a céget abban, hogy könnyedén nyomonkövethesse az egyes részlegekre érkező megrendeléseket és azok státuszát. Továbbá a részlegeken dolgozókat és beosztásaikat is számontartja; az adott részleg vezetőjének információja, beleértve a vezetői státuszának megkezdési dátumát, is egyaránt eltárolása kerül a többi dolgozó adatai mellett.

# 1. Feladat

## a) Az adatbázis ER modell tervezése

Az alábbi felsorolás jellemzi az ER modellben található egyes egyedeket és azoknak tulajdonságait részletesen. Az adatbázis 4 egyeddel rendelkezik, ezen felül a kapcsolatok összes típusából számos megtalálható az egyedek között, ezek névszerint az 1:1, 1:N, valamint az N:N kapcsolatok.

Egy kapcsolat kivételével az összes kapcsolat rendelkezik valamilyen fajta tulajdonsággal.

Az adatbázis egyedei részletesen:

### **Rendelés**

- Rid (Egyéni kulcs, rendelés azonosítója)
- Státusz (Logikai érték, mutatja, hogy megérkezett-e az áru)
- Termék név (A rendelt áru teljes neve)
- Szállítási hely (A leszállítandó áru célja)
  - Irsz. (Irányítószám)
  - Város
  - Utca, házszám
- Mennyiség (A rendelt termék pontos mértéke)
- Ár (A rakomány teljes költsége, mely kiszámítható az egyes áruk mértékének és árának összegeként)

### **Megrendelő**

- Mid (Egyéni azonosító)

- Cégnév (A rendelést leadó cég neve)
- Elérhetőség (Rendelő elérhetőségi lehetőségeinek listája)
  - Tel. Szám (Telefon szám, akár több is)
  - Email (Érvényes email cím)

### **Részleg**

- Rid (Egyéni kulcs, azonosító)
- Név (Az adott részleg neve, dezinációja)
- Dolgozók száma (Kiszámítható a dolgozik kapcsolat és a dolgozók tábla felhasználásával)

### **Dolgozó**

- Did (dolgozó egyéni azonosítója)
- Név (Dolgozó neve)

Kapcsolatok:

### **Rendel**

- Az egyes megrendelőket köti össze az általuk leadott rendelések táblájával
- Tartalmazza a rendelés dátumát

### **Érkezik**

- Adott részlegekre érkeznek megrendelések

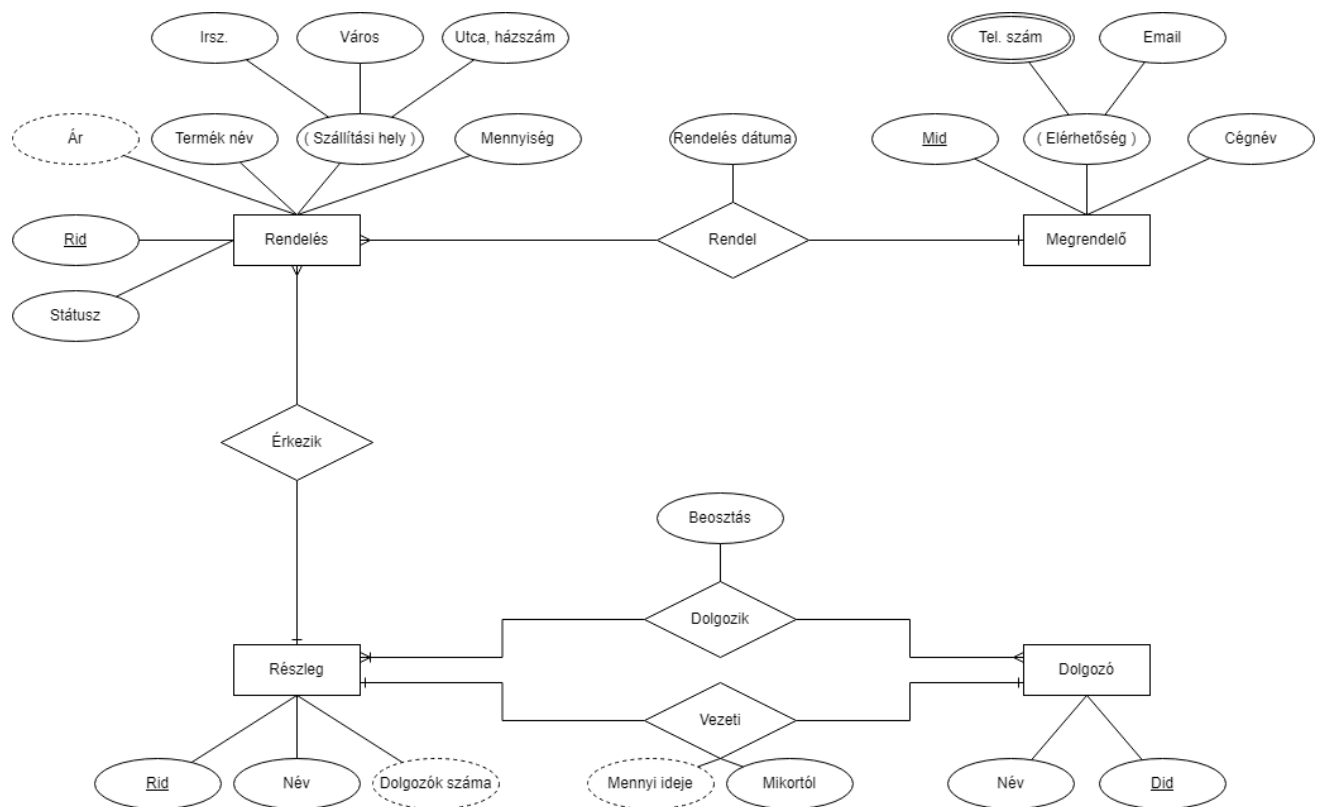
### **Vezeti**

- Minden részleget vezet egy, a részlegen dolgozók közül, kijelölt személy

- Tartalmazza a vezetési pozíció felvételének idejét
- Kiszámítható, hogy az illető mennyi ideje vezeti a kérdéses részleget

## Dolgozik

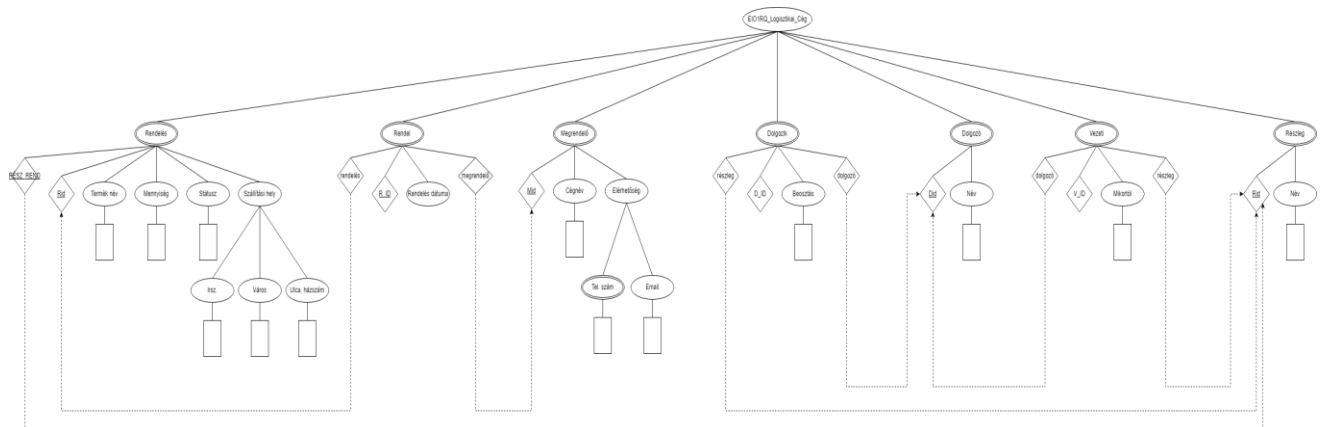
- Az egyes részlegeket köti össze a dolgozókkal, így számon tartva őket részlegekre bontva
- Minden dolgozónak szerepel az adatbázisban a részlegére vonatkozó munkakri beosztás



## b) Az adatbázis konvertálása XDM modellre

Az ER modell XDM modellre konvertálása során az összes egyed, ebbe a kapcsolatokat is beleértve az XDM modell gyökérelemének közvetlen gyermekeivé válnak.

Az átalakítás során az egyedek közti kapcsolat jelölése szaggatott vonal segítségével történik, melynek konstrukciója folyamán jelentős figyelmet kell fordítani, hogy ezek ne keresztezzék egymást.



### c) Az XDM modell alapján XML dokumentum készítése

Az XDM modellről való átkonvertálás során az idegen illetve elsődleges kulcsok attribútumként helyezkednek el. Az alábbi XML dokumentum minden egyedből legalább 3 példányt tartalmaz.

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <EIO1RQ_Logisztikai_Cég xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
4  xs:noNamespaceSchemaLocation="XMLSchemaeio1rq.xsd">
5
6      <!-- Rendelés -->
7      <rendelés RENDELÉS_ID="1" RÉSZLEG_FK="1">
8          <termék_név>Ásványvíz</termék_név>
9          <menyiség>13</menyiség>
10         <leszállítva>true</leszállítva>
11         <szállítási_hely>
12             <irányító_száma>3519</irányító_száma>
13             <város>Miskolc</város>
14             <utca_háza>Apat utca 12.</utca_háza>
15         </szállítási_hely>
16     </rendelés>
17
18     <rendelés RENDELÉS_ID="2" RÉSZLEG_FK="2">
19         <termék_név>Hamburger pogácsa</termék_név>
20         <menyiség>20</menyiség>
21         <leszállítva>false</leszállítva>
22         <szállítási_hely>
23             <irányító_száma>6723</irányító_száma>
24             <város>Szedeg</város>
25             <utca_háza>Eperjesi sor 2.</utca_háza>
26         </szállítási_hely>
27     </rendelés>
28
29     <rendelés RENDELÉS_ID="3" RÉSZLEG_FK="3">
30         <termék_név>Sony WH-XB910N fejhallgató</termék_név>
31         <menyiség>8</menyiség>
32         <leszállítva>true</leszállítva>
33         <szállítási_hely>
34             <irányító_száma>9025</irányító_száma>
35             <város>Győr</város>
36             <utca_háza>Akac utca 35.</utca_háza>
37         </szállítási_hely>
38     </rendelés>
39
40     <rendelés RENDELÉS_ID="4" RÉSZLEG_FK="2">
41         <termék_név>Sertéskaraj</termék_név>
42         <menyiség>20</menyiség>

```



```

41     <termék_név>Sertéskaraj</termék_név>
42     <menyiség>20</menyiség>
43     <leszállítva>true</leszállítva>
44     <szállítási_hely>
45         <irányító_száma>7634</irányító_száma>
46         <város>Pécs</város>
47         <utca_házszám>Lankas utca 22.</utca_házszám>
48     </szállítási_hely>
49 </rendelés>
50
51 <rendelés RENDELÉS_ID="5" RÉSZLEG_FK="1">
52     <termék_név>Coca Cola Cherry 6x1.5L</termék_név>
53     <menyiség>17</menyiség>
54     <leszállítva>false</leszállítva>
55     <szállítási_hely>
56         <irányító_száma>4028</irányító_száma>
57         <város>Debrecen</város>
58         <utca_házszám>Abonyi utca 72.</utca_házszám>
59     </szállítási_hely>
60 </rendelés>
61
62 <!-- Rendel -->
63 <rendel RENDEL_ID="1" RENDELÉS_FK="1" MEGRENDELŐ_FK="1">
64     <rendelés_dátuma>2023-01-17</rendelés_dátuma>
65 </rendel>
66
67 <rendel RENDEL_ID="2" RENDELÉS_FK="2" MEGRENDELŐ_FK="2">
68     <rendelés_dátuma>2023-06-04</rendelés_dátuma>
69 </rendel>
70
71 <rendel RENDEL_ID="3" RENDELÉS_FK="3" MEGRENDELŐ_FK="3">
72     <rendelés_dátuma>2023-08-23</rendelés_dátuma>
73 </rendel>
74
75 <rendel RENDEL_ID="4" RENDELÉS_FK="4" MEGRENDELŐ_FK="1">
76     <rendelés_dátuma>2023-02-10</rendelés_dátuma>
77 </rendel>
78
79 <rendel RENDEL_ID="5" RENDELÉS_FK="5" MEGRENDELŐ_FK="3">
80     <rendelés_dátuma>2022-11-03</rendelés_dátuma>
81 </rendel>
82

```

```

80         <rendeles_datuma>2022-11-03</rendeles_datuma>
81     </rendel>
82
83     <!-- Megrendelő -->
84     <megrendelő MEGRENDELŐ_ID="1">
85         <cégnév>Fenyő és Társa Kft.</cégnév>
86         <elérhetőség>
87             <tel_szám>+3612045966</tel_szám>
88             <email>info@tlx.hu</email>
89         </elérhetőség>
90     </megrendelő>
91
92     <megrendelő MEGRENDELŐ_ID="2">
93         <cégnév>Rothmayer Bt.</cégnév>
94         <elérhetőség>
95             <tel_szám>+3633413665</tel_szám>
96             <email>rothmayer1@gmail.com</email>
97         </elérhetőség>
98     </megrendelő>
99
100    <megrendelő MEGRENDELŐ_ID="3">
101        <cégnév>Horváth Uno Kft.</cégnév>
102        <elérhetőség>
103            <tel_szám>+36705469391</tel_szám>
104            <email>horvathuno@gmail.com</email>
105        </elérhetőség>
106    </megrendelő>
107
108
109
110    <!-- Dolgozik -->
111    <dolgozik DOLGOZIK_ID="1" RÉSZLEG_FK="1" DOLGOZÓ_FK="1">
112        <beosztás>Logisztikai menedzser</beosztás>
113    </dolgozik>
114
115    <dolgozik DOLGOZIK_ID="2" RÉSZLEG_FK="1" DOLGOZÓ_FK="2">
116        <beosztás>Élelmiszer mérnök</beosztás>
117    </dolgozik>
118
119    <dolgozik DOLGOZIK_ID="3" RÉSZLEG_FK="2" DOLGOZÓ_FK="3">
120        <beosztás>Targonca kezelő</beosztás>
121    </dolgozik>

```

```

119 <dolgozik DOLGOZIK_ID="3" RÉSZLEG_FK="2" DOLGOZÓ_FK="3">
120   <beosztás>Targonca kezelő</beosztás>
121 </dolgozik>
122
123 <dolgozik DOLGOZIK_ID="4" RÉSZLEG_FK="2" DOLGOZÓ_FK="4">
124   <beosztás>Villanyszerelő</beosztás>
125 </dolgozik>
126
127 <dolgozik DOLGOZIK_ID="5" RÉSZLEG_FK="3" DOLGOZÓ_FK="5">
128   <beosztás>Informatikus</beosztás>
129 </dolgozik>
130
131 <dolgozik DOLGOZIK_ID="6" RÉSZLEG_FK="3" DOLGOZÓ_FK="6">
132   <beosztás>Gazdasági tanácsadó</beosztás>
133 </dolgozik>
134
135
136
137 <!-- Dolgozó -->
138 <dolgozó DOLGOZÓ_ID="1">
139   <név>Kovács Béla</név>
140 </dolgozó>
141
142 <dolgozó DOLGOZÓ_ID="2">
143   <név>Kiss János</név>
144 </dolgozó>
145
146 <dolgozó DOLGOZÓ_ID="3">
147   <név>Eged Csanád</név>
148 </dolgozó>
149
150 <dolgozó DOLGOZÓ_ID="4">
151   <név>Kun Tamás</név>
152 </dolgozó>
153
154 <dolgozó DOLGOZÓ_ID="5">
155   <név>Jámbor Andrea</név>
156 </dolgozó>
157
158 <dolgozó DOLGOZÓ_ID="6">
159   <név>Tóth Sándor</név>

```

```

153
154     <dolgozó DOLGOZÓ_ID="5">
155         <név>Jámbor Andrea</név>
156     </dolgozó>
157
158     <dolgozó DOLGOZÓ_ID="6">
159         <név>Tóth Sándor</név>
160     </dolgozó>
161
162
163
164     <!-- Vezeti -->
165     <vezeti VEZETI_ID="1" RÉSZLEG_FK="1" DOLGOZÓ_FK="1">
166         <mikortól>2017-01-01</mikortól>
167     </vezeti>
168
169     <vezeti VEZETI_ID="2" RÉSZLEG_FK="2" DOLGOZÓ_FK="3">
170         <mikortól>2019-04-15</mikortól>
171     </vezeti>
172
173     <vezeti VEZETI_ID="3" RÉSZLEG_FK="3" DOLGOZÓ_FK="6">
174         <mikortól>2020-09-07</mikortól>
175     </vezeti>
176
177
178
179     <!-- Részleg -->
180     <részleg RÉSZLEG_ID="1">
181         <név>Élelmiszer</név>
182     </részleg>
183
184     <részleg RÉSZLEG_ID="2">
185         <név>Ital</név>
186     </részleg>
187
188     <részleg RÉSZLEG_ID="3">
189         <név>Informatika</név>
190     </részleg>
191
192 </EI01RQ_Logisztikai_Cég>

```

d) Az XML dokumentum alapján XMLSchema készítése- saját típusok, ref, key, keyref, speciális elemek

Az XML dokumentumhoz tartozó XMLSchema elkészítése során első lépésként az egyszerű, illetve saját típusokat kigyűjtjük, majd a komplex típusokat és végül a kucsokat.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
4
5      <!-- Egyszerű típusok kigyűjtése -->
6      <xs:element name="irányító_száma" type="irszámTípus"/>
7      <xs:element name="város" type="xs:string"/>
8      <xs:element name="utca_háza" type="utcaházaTípus"/>
9      <xs:element name="termék_név" type="xs:string"/>
10     <xs:element name="mennyiség" type="mennyiségTípus"/>
11     <xs:element name="leszállítva" type="xs:boolean"/>
12     <xs:element name="rendelés_dátuma" type="xs:date"/>
13     <xs:element name="tel_száma" type="telefonTípus"/>
14     <xs:element name="email" type="emailTípus"/>
15     <xs:element name="cégnév" type="xs:string"/>
16     <xs:element name="beosztás" type="xs:string"/>
17     <xs:element name="mikortól" type="xs:date"/>
18     <xs:element name="név" type="xs:string"/>
19
20
21     <!-- Egyszerű típusok definiálása -->
22     <xs:simpleType name="irszámTípus">
23         <xs:restriction base="xs:int">
24             <xs:minInclusive value="1000" />
25             <xs:maxInclusive value="9999" />
26         </xs:restriction>
27     </xs:simpleType>
28
29     <xs:simpleType name="utcaházaTípus">
30         <xs:restriction base="xs:string">
31             <xs:pattern value="([a-z A-Z]* \d+.( \d+\/\d+)?)" />
32         </xs:restriction>
33     </xs:simpleType>
34
35     <xs:simpleType name="mennyiségTípus">
36         <xs:restriction base="xs:int">
37             <xs:minInclusive value="1" />
38         </xs:restriction>
39     </xs:simpleType>
40
41     <xs:simpleType name="telefonTípus">
42         <xs:restriction base="xs:string">
```

```

40
41 <xs:simpleType name="telefonTípus">
42   <xs:restriction base="xs:string">
43     <xs:pattern value="(\+36\ ?\d{1,2}\ ?\d{3}\ ?\d{4})" />
44   </xs:restriction>
45 </xs:simpleType>
46
47 <xs:simpleType name="emailTípus">
48   <xs:restriction base="xs:string">
49     <xs:pattern value="([a-zA-Z0-9]*@[a-zA-Z]*.[a-zA-Z]*)" />
50   </xs:restriction>
51 </xs:simpleType>
52
53
54
55 <!-- Komplex típusok definiálása-->
56 <xs:complexType name="rendelésTípus">
57   <xs:sequence>
58     <xs:element ref="termék_név"/>
59     <xs:element ref="mennyiség"/>
60     <xs:element ref="leszállítva"/>
61     <xs:element name="szállítási_hely">
62       <xs:complexType>
63         <xs:sequence>
64           <xs:element ref="irányító_szám"/>
65           <xs:element ref="város"/>
66           <xs:element ref="utca_házzsám"/>
67         </xs:sequence>
68       </xs:complexType>
69     </xs:element>
70   </xs:sequence>
71   <xs:attribute name="RENDELÉS_ID" type="xs:integer" use="required" />
72   <xs:attribute name="RÉSZLEG_FK" type="xs:integer" use="required" />
73 </xs:complexType>
74
75 <xs:complexType name="rendelTípus">
76   <xs:sequence>
77     <xs:element ref="rendelés_dátuma"/>
78   </xs:sequence>
79   <xs:attribute name="RENDEL_ID" type="xs:integer" use="required" />
80   <xs:attribute name="RENDELÉS_FK" type="xs:integer" use="required" />
81   <xs:attribute name="MEGRENDELŐ_FK" type="xs:integer" use="required" />

```

```

80     <xs:attribute name="RENDELÉS_FK" type="xs:integer" use="required" />
81     <xs:attribute name="MEGRENDELŐ_FK" type="xs:integer" use="required" />
82 </xs:complexType>
83
84 <xs:complexType name="megrendelőTípus">
85     <xs:sequence>
86         <xs:element ref="cégnév"/>
87         <xs:element name="elérhetőség">
88             <xs:complexType>
89                 <xs:sequence>
90                     <xs:element ref="tel_szám"/>
91                     <xs:element ref="email"/>
92                 </xs:sequence>
93             </xs:complexType>
94         </xs:element>
95     </xs:sequence>
96     <xs:attribute name="MEGRENDELŐ_ID" type="xs:integer" use="required" />
97 </xs:complexType>
98
99 <xs:complexType name="dolgozikTípus">
100     <xs:sequence>
101         <xs:element ref="beosztás"/>
102     </xs:sequence>
103     <xs:attribute name="DOLGOZIK_ID" type="xs:integer" use="required" />
104     <xs:attribute name="RÉSZLEG_FK" type="xs:integer" use="required" />
105     <xs:attribute name="DOLGOZÓ_FK" type="xs:integer" use="required" />
106 </xs:complexType>
107
108 <xs:complexType name="dolgozóTípus">
109     <xs:sequence>
110         <xs:element ref="név"/>
111     </xs:sequence>
112     <xs:attribute name="DOLGOZÓ_ID" type="xs:integer" use="required" />
113 </xs:complexType>
114
115 <xs:complexType name="vezetiTípus">
116     <xs:sequence>
117         <xs:element ref="mikortól"/>
118     </xs:sequence>
119     <xs:attribute name="VEZETI_ID" type="xs:integer" use="required" />
120     <xs:attribute name="DOLGOZÓ_FK" type="xs:integer" use="required" />
121     <xs:attribute name="RÉSZLEG_FK" type="xs:integer" use="required" />

```

```

120 </xs:attribute name="RÉSZLEG_FK" type="xs:integer" use="required" />
121 <xs:attribute name="RÉSZLEG_FK" type="xs:integer" use="required" />
122 </xs:complexType>
123
124 <xs:complexType name="részlegTípus">
125 <xs:sequence>
126 <xs:element ref="név"/>
127 </xs:sequence>
128 <xs:attribute name="RÉSZLEG_ID" type="xs:integer" use="required" />
129 </xs:complexType>
130
131
132
133 <!-- Dokumentum felépítése a gyökérleemtől -->
134 <xs:element name="EIO1RQ_Logisztikai_Cég">
135 <xs:complexType>
136 <xs:sequence>
137 <xs:element name="rendelés" type="rendelésTípus" minOccurs="0" maxOccurs="unbounded"/>
138 <xs:element name="rendel" type="rendelTípus" minOccurs="0" maxOccurs="unbounded"/>
139 <xs:element name="megrendelő" type="megrendelőTípus" minOccurs="0" maxOccurs="10"/>
140 <xs:element name="dolgozik" type="dolgozikTípus" minOccurs="0" maxOccurs="unbounded"/>
141 <xs:element name="dolgozó" type="dolgozóTípus" minOccurs="0" maxOccurs="unbounded"/>
142 <xs:element name="vezeti" type="vezetiTípus" minOccurs="0" maxOccurs="unbounded"/>
143 <xs:element name="részleg" type="részlegTípus" minOccurs="0" maxOccurs="20"/>
144 </xs:sequence>
145 </xs:complexType>
146
147
148
149 <!-- Elsődleges kulcsok-->
150 <xs:key name="rendelés_kulcs">
151 <xs:selector xpath="rendelés"/>
152 <xs:field xpath="@RENDELÉS_ID"/>
153 </xs:key>
154
155 <xs:key name="rendel_kulcs">
156 <xs:selector xpath="rendel"/>
157 <xs:field xpath="@RENDEL_ID"/>
158 </xs:key>
159
160 <xs:key name="megrendelő_kulcs">
161 <xs:selector xpath="megrendelő"/>

```



```

159
160 <xs:key name="megrendelő_kulcs">
161   <xs:selector xpath="megrendelő"/>
162   <xs:field xpath="@MEGRENDELŐ_ID"/>
163 </xs:key>
164
165 <xs:key name="dolgozik_kulcs">
166   <xs:selector xpath="dolgozik"/>
167   <xs:field xpath="@DOLGOZIK_ID"/>
168 </xs:key>
169
170 <xs:key name="dolgozó_kulcs">
171   <xs:selector xpath="dolgozó"/>
172   <xs:field xpath="@DOLGOZÓ_ID"/>
173 </xs:key>
174
175 <xs:key name="vezeti_kulcs">
176   <xs:selector xpath="vezeti"/>
177   <xs:field xpath="@VEZETI_ID"/>
178 </xs:key>
179
180 <xs:key name="részleg_kulcs">
181   <xs:selector xpath="részleg"/>
182   <xs:field xpath="@RÉSZLEG_ID"/>
183 </xs:key>
184
185
186
187 <!-- Idegen kulcsok-->
188 <xs:keyref name="részleg_rendelés_kulcs" refer="részleg_kulcs">
189   <xs:selector xpath="rendelés"/>
190   <xs:field xpath="@részleg"/>
191 </xs:keyref>
192
193 <xs:keyref name="rendel_rendelés_kulcs" refer="rendelés_kulcs">
194   <xs:selector xpath="rendel"/>
195   <xs:field xpath="@rendelés"/>
196 </xs:keyref>
197
198 <xs:keyref name="rendel_megrendelő_kulcs" refer="megrendelő_kulcs">
199   <xs:selector xpath="rendel"/>
200   <xs:field xpath="@megrendelő"/>

```

Line 225, Column 13

```

207
208     <xs:keyref name="dolgozik_dolgozó_kulcs" refer="dolgozó_kulcs">
209         <xs:selector xpath="dolgozik"/>
210         <xs:field xpath="@dolgozó"/>
211     </xs:keyref>
212
213     <xs:keyref name="vezeti_dolgozó_kulcs" refer="dolgozó_kulcs">
214         <xs:selector xpath="vezeti"/>
215         <xs:field xpath="@dolgozó"/>
216     </xs:keyref>
217
218     <xs:keyref name="vezeti_részleg_kulcs" refer="részleg_kulcs">
219         <xs:selector xpath="vezeti"/>
220         <xs:field xpath="@részleg"/>
221     </xs:keyref>
222
223 </xs:element>
224
225 </xs:schema>

```

## 2. Feladat

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

### 2a) Adatolvasás

Az alábbi program beolvassa a kijelölt XML fájlt, majd kiírja annak struktúrált tartalmát a konzolra , valamint egy külső fájlba is.

```

1 package hu.domparse.eio1rq;
2
3 import java.io.File;
4 import java.io.FileWriter;
5 import java.util.ArrayList;
6
7 import javax.xml.parsers.DocumentBuilder;
8 import javax.xml.parsers.DocumentBuilderFactory;
9
10 import org.w3c.dom.Document;
11 import org.w3c.dom.Node;
12 import org.w3c.dom.NodeList;
13
14 public class DomReadEIO1RQ {
15     public static void main(String args[]) {
16         System.out.println("2a) Adatolvasás");
17
18         try {
19             // File létrehozása és előkészítés a feldolgozásra
20             File inputFile = new File("XMLeio1rq.xml");
21             DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
22             factory.setNamespaceAware(true);
23             DocumentBuilder builder = factory.newDocumentBuilder();
24             Document doc = builder.parse(inputFile);
25
26             // A gyökérelem minden gyerekelemének elkülönítése
27             NodeList root = doc.getChildNodes();
28             // DOM fa leveleinek feltérképezése
29             ArrayList<String> content = getAllLeaves((Node)root);
30
31             // Fa struktúrált kiírása konzolra és fájlba
32             FileWriter writer = new FileWriter("XMLeio1rq_Szukturalt.txt");
33             for(int i = 0; i < content.size(); i++) {
34                 System.out.print(content.get(i));
35                 writer.write(content.get(i));
36             }
37             writer.close();
38
39         } catch (Exception e) {
40             e.printStackTrace();
41         }
42     }
43
44     // DOM fa leveleinek rekurzív bejárása
45     public static ArrayList<String> getAllLeaves(Node node) {
46         ArrayList<String> content = new ArrayList<String>();
47         NodeList children = node.getChildNodes();
48         if(children.getLength() == 0) {
49             if(node.getNodeType() != Node.COMMENT_NODE) content.add(" "+node.getTextContent());
50         }
51     }

```

```

45 // DOM fa leveleinek rekurzív bejárása
46 public static ArrayList<String> getAllLeaves(Node node) {
47     ArrayList<String> content = new ArrayList<String>();
48     NodeList children = node.getChildNodes();
49     if(children.getLength() == 0) {
50         if(node.getNodeType() != Node.COMMENT_NODE) content.add(" "+node.getTextContent());
51     }
52     else {
53         // Elágazások hozzáadása a megfelelő tartalom szeparáció érdekében
54         if(node.getNodeType() != Node.DOCUMENT_NODE) content.add(node.getNodeName());
55         int len = children.getLength();
56         // Gyökérelemet leszámítva a fa minden szintjén elhagyjuk az utolsó üres #text elemeti
57         // Szebb indentálás érdekében
58         len -= (len > 1) ? 1 : 0;
59         for(int i = 0; i < len; i++) {
60             content.addAll(getAllLeaves(children.item(i)));
61         }
62     }
63     return content;
64 }
65 }
66 }
67

```

2b) Adatmódosítás (kód - comment együtt) fájlnev:

DOMModifyNeptunkod.java

A következő java program a feladatban szereplő XML fájlban 5 módosítást végez el.

```

1 package hu.domparse.eio1rq;
2
3 import java.io.File;
4 import java.text.SimpleDateFormat;
5 import java.util.Date;
6
7 import javax.xml.parsers.DocumentBuilder;
8 import javax.xml.parsers.DocumentBuilderFactory;
9 import javax.xml.transform.Transformer;
10 import javax.xml.transform.TransformerFactory;
11 import javax.xml.transform.dom.DOMSource;
12 import javax.xml.transform.stream.StreamResult;
13
14 import org.w3c.dom.Document;
15 import org.w3c.dom.Element;
16 import org.w3c.dom.NodeList;
17
18 public class DomModifyEIO1RQ {
19     public static void main(String args[]) {
20         try {
21             File inputFile = new File("XMLeio1rq.xml");
22
23             // Document létrehozása
24             DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
25             DocumentBuilder domBuilder = dbFactory.newDocumentBuilder();
26             Document doc = domBuilder.parse(inputFile);
27
28             // Kiválasztott elemek módosítása
29             modify(doc);
30
31             // Új dokumentum mentése
32             TransformerFactory transformerFactory = TransformerFactory.newInstance();
33             Transformer transformer = transformerFactory.newTransformer();
34             DOMSource source = new DOMSource(doc);
35             StreamResult result = new StreamResult(new File("XMLeio1rq_Modified.xml"));
36             transformer.transform(source, result);
37
38             // Konzolra kiíratás
39             StreamResult resultConsole = new StreamResult(System.out);
40             transformer.transform(source, resultConsole);
41         } catch (Exception e) {
42             e.printStackTrace();
43         }
44     }
45
46     public static void modify(Document doc) {
47         // 4. Rendelés nevének módosítása
48         NodeList orders = doc.getElementsByTagName("rendelés");
49         Element order = (Element) orders.item(3);
50         order.getElementsByTagName("termék_név").item(0).setTextContent("Csirkemellfilé");
51     }

```

```

39     StreamResult resultConsole = new StreamResult(System.out);
40     transformer.transform(source, resultConsole);
41 } catch(Exception e) {
42     e.printStackTrace();
43 }
44 }
45
46 public static void modify(Document doc) {
47     // 4. Rendelés nevének módosítása
48     NodeList orders = doc.getElementsByTagName("rendelés");
49     Element order = (Element) orders.item(3);
50     order.getElementsByTagName("termék_név").item(0).setTextContent("Csirkemellfilé");
51
52     // 2. Megrendelő email címének módosítása
53     NodeList clients = doc.getElementsByTagName("megrendelő");
54     Element client = (Element) clients.item(1);
55     client.getElementsByTagName("elérhetőség").item(0).setTextContent("rothmayerbt@gmail.com");
56
57     // 4. Dolgozó nevének megváltoztatása
58     NodeList workers = doc.getElementsByTagName("dolgozó");
59     Element worker = (Element) workers.item(4);
60     worker.getElementsByTagName("név").item(0).setTextContent("Cziglédi Bálint");
61
62     // 1. Részleg vezetőjének és a kinevezés dátumának módosítása
63     NodeList leaders = doc.getElementsByTagName("vezeti");
64     Element leader = (Element) leaders.item(0);
65     leader.setAttribute("DOLGOZÓ_FK", "5");
66     leader.getElementsByTagName("mikortól").item(0).setTextContent(new SimpleDateFormat("yyyy-MM-dd").format(new Date()));
67
68     // Legrégábbi rendelés dátumának módosítás
69     NodeList orderList = doc.getElementsByTagName("rendel");
70     Element orderElement = (Element) orderList.item(4);
71     orderElement.getElementsByTagName("rendelés_dátuma").item(0).setTextContent(new SimpleDateFormat("yyyy-MM-dd").format(new Date()));
72
73 }
74 }
75

```

## 2c) Adatlekérdezés

A lekérdezés során kinyerni kívánt adatok az azokat követő kód elején található és tájékoztatást adnak, hogy mely információkkal fogunk későbbiekben dolgozni.

```

1 package hu.domparse.eio1rq;
2
3 import javax.xml.parsers.DocumentBuilderFactory;
4
13
14 public class DomQueryEIO1RQ {
15     public static void main(String args[]) {
16         try {
17             // XML fájl beolvasása és dokumentum objekt előkészítése
18             File inputFile = new File("XMLeio1rq.xml");
19             DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
20             DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
21             Document doc = dBuilder.parse(inputFile);
22             doc.getDocumentElement().normalize();
23
24             // Gyökérelem azonosítása
25             System.out.println("2b) Gyökérelem: ");
26             System.out.println(doc.getDocumentElement().getNodeName());
27             NodeList nodes = doc.getElementsByTagName("rendelés");
28             System.out.println("-----");
29
30             // A rendelések kilistázása
31             System.out.println("\nAz összes rendelés listája");
32             System.out.println("-----");
33             for(int i = 0; i < nodes.getLength(); i++) {
34                 Node node = nodes.item(i);
35                 if (node.getNodeType() == Node.ELEMENT_NODE) {
36                     Element eElement = (Element) node;
37                     System.out.println(eElement.getElementsByTagName("termék_név").item(0).getTextContent());
38                 }
39             }
40
41             // A cégnél található részlegek száma
42             System.out.println("\nRészlegek száma");
43             System.out.println("-----");
44             nodes = doc.getElementsByTagName("részleg");
45             System.out.println(nodes.getLength());
46
47
48             // Egyes részlegekre érkezett megrendelések száma
49             System.out.println("\nMelyik részlegre mennyi megrendelés érkezett");
50             System.out.println("-----");
51             int[] rendelések = new int[nodes.getLength()];
52
53             nodes = doc.getElementsByTagName("rendelés");
54             for(int i = 0; i < nodes.getLength(); i++) {
55                 Node node = nodes.item(i);
56                 if (node.getNodeType() == Node.ELEMENT_NODE) {
57                     Element eElement = (Element) node;
58                     rendelések[Integer.parseInt(eElement.getAttribute("RÉSZLEG_FK")) - 1]++;
59                 }

```



```

55         Node node = nodes.item(i);
56         if (node.getNodeType() == Node.ELEMENT_NODE) {
57             Element eElement = (Element) node;
58             rendelesek[Integer.parseInt(eElement.getAttribute("RÉSZLEG_FK")) - 1]++;
59         }
60     }
61     nodes = doc.getElementsByTagName("részleg");
62     for(int i = 0; i < nodes.getLength(); i++) {
63         Node node = nodes.item(i);
64         if (node.getNodeType() == Node.ELEMENT_NODE) {
65             Element eElement = (Element) node;
66             System.out.println(eElement.getElementsByTagName("név").item(0).getTextContent() + ": " + rendelesek[
67         }
68     }
69
70     // Kilistázzuk a kizárólag 2. részlegen dolgozók neveit
71     System.out.println("\nA második részlegen dolgozók listája");
72     System.out.println("-----");
73     nodes = doc.getElementsByTagName("dolgozik");
74     List<String> workersID = new ArrayList<String>();
75     for(int i = 0; i < nodes.getLength(); i++) {
76         Node node = nodes.item(i);
77         if (node.getNodeType() == Node.ELEMENT_NODE) {
78             Element eElement = (Element) node;
79             if(eElement.getAttribute("RÉSZLEG_FK").equals("2")) {
80                 workersID.add(eElement.getAttribute("DOLGOZÓ_FK"));
81             }
82         }
83     }
84     nodes = doc.getElementsByTagName("dolgozó");
85     for(int i = 0; i < nodes.getLength(); i++) {
86         Node node = nodes.item(i);
87         if (node.getNodeType() == Node.ELEMENT_NODE) {
88             Element eElement = (Element) node;
89             if(workersID.contains(eElement.getAttribute("DOLGOZÓ_ID"))) {
90                 System.out.println(eElement.getElementsByTagName("név").item(0).getTextContent());
91             }
92         }
93     }
94
95     // Bizonyos részlegek vezetői mikor szereték meg a pozíciót
96     System.out.println("\nEgyes részlegeket mikortól vezeti a kijelölt személy");
97     System.out.println("-----");
98     nodes = doc.getElementsByTagName("vezeti");
99     NodeList workers = doc.getElementsByTagName("dolgozó");
100    NodeList divs = doc.getElementsByTagName("részleg");
101    for(int i = 0; i < nodes.getLength(); i++) {
102        Node node = nodes.item(i);
103        if (node.getNodeType() == Node.ELEMENT_NODE) {
104            Element eElement = (Element) node;

```

```

94
95 // Bizonyos részlegek vezetői mikor szereték meg a pozíciót
96 System.out.println("\nEgyes részlegeket mikortól vezeti a kijelölt személy");
97 System.out.println("-----");
98 nodes = doc.getElementsByTagName("vezeti");
99 NodeList workers = doc.getElementsByTagName("dolgozó");
100 NodeList divs = doc.getElementsByTagName("részleg");
101 for(int i = 0; i < nodes.getLength(); i++) {
102     Node node = nodes.item(i);
103     if (node.getNodeType() == Node.ELEMENT_NODE) {
104         Element eElement = (Element) node;
105         eElement.getAttribute("RÉSZLEG_FK");
106
107         for(int j = 0; j < divs.getLength(); j++) {
108             Node n1 = divs.item(j);
109             if (n1.getNodeType() == Node.ELEMENT_NODE) {
110                 Element elem1 = (Element) n1;
111                 if(elem1.getAttribute("RÉSZLEG_ID").equals(eElement.getAttribute("RÉSZLEG_FK"))) {
112                     System.out.print(elem1.getElementsByTagName("név").item(0).getTextContent() + " : ")
113                 }
114             }
115         }
116
117         for(int j = 0; j < workers.getLength(); j++) {
118             Node n1 = workers.item(j);
119             if (n1.getNodeType() == Node.ELEMENT_NODE) {
120                 Element elem1 = (Element) n1;
121                 if(elem1.getAttribute("DOLGOZÓ_ID").equals(eElement.getAttribute("DOLGOZÓ_FK"))) {
122                     System.out.println(elem1.getElementsByTagName("név").item(0).getTextContent());
123                 }
124             }
125         }
126     }
127 }
128 }
129 }catch(Exception e) {
130     e.printStackTrace();
131 }
132 }
133 }
134

```

2d) Adatírás -készítsen egy DOM API programot, amely egy XMLNeptunkod.xml dokumentum tartalmát fa struktúra formában kiírja a konzolra és egy XMLNeptunkod1.xml fájlba.(kód –comment együtt)–fájlnév: DOMWriteNeptunkod.java

A soron következő program felépít egy DOM fát a megadott adatok segítségével, mely egyben meghatározza a dokumentum struktúráját is. Majd ebből a fából generál egy XML fájlt a megfelelő adatokkal.

```

1 package hu.dompars.eio1rq;
2
3
4 import java.io.File;
18 |
19 public class DomWriteEIO1RQ {
20     public static void main(String args[]) {
21         try {
22             DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
23             DocumentBuilder dbBuilder = dbFactory.newDocumentBuilder();
24             Document document = dbBuilder.newDocument();
25
26             // Gyökérelem
27             Element root = document.createElement("EIO1RQ_Logisztikai_Cég");
28             root.setAttribute("xmlns:xs", "http://www.w3.org/2001/XMLSchema-instance");
29             root.setAttribute("xs:noNamespaceSchemaLocation", "XMLSchemaeio1rq.xsd");
30             document.appendChild(root);
31
32             // Rendelés
33             root.appendChild(document.createComment("Rendelések"));
34             createOrder(document, root, "1", "1", "Ásványvíz", "13", "true", "3519", "Miskolc", "Apat utca 12.");
35             createOrder(document, root, "2", "2", "Hamburger pogácsa", "20", "false", "6723", "Szeged", "Eperjesi sor 2.");
36             createOrder(document, root, "3", "3", "Sony WH-XB910N fejhallgató", "8", "true", "9025", "Győr", "Akac utca 35.");
37             createOrder(document, root, "4", "4", "Sertéskaraj", "20", "true", "7634", "Pécs", "Lankas utca 22.");
38             createOrder(document, root, "5", "5", "Coca Cola Cherry 6x1.5L", "17", "false", "4028", "Debrecen", "Abonyi utca 72.");
39
40             // Rendel kapcsolat
41             root.appendChild(document.createComment("Rendel"));
42             createOrderConn(document, root, "1", "1", "1", "2023-01-17");
43             createOrderConn(document, root, "2", "2", "2", "2023-06-04");
44             createOrderConn(document, root, "3", "3", "3", "2023-08-23");
45             createOrderConn(document, root, "4", "4", "1", "2023-02-10");
46             createOrderConn(document, root, "5", "5", "3", "2022-11-03");
47
48
49             // Megrendelő
50             root.appendChild(document.createComment("Megrendelő"));
51             createClient(document, root, "1", "Fenyő és Társa Kft.", "+3612045966", "info@tlx.hu");
52             createClient(document, root, "2", "Rothmayer Bt.", "+3633413665", "rothmayer1@gmail.com");
53             createClient(document, root, "3", "Horváth Uno Kft.", "+36705469391", "horvathuno@gmail.com");
54
55             // Dolgozik kapcsolat
56             root.appendChild(document.createComment("Dolgozik"));
57             createWorking(document, root, "1", "1", "1", "Logisztikai menedzser");
58             createWorking(document, root, "2", "1", "2", "Logisztikai menedzser");
59             createWorking(document, root, "3", "2", "3", "Logisztikai menedzser");
60             createWorking(document, root, "4", "2", "4", "Logisztikai menedzser");
61             createWorking(document, root, "5", "3", "5", "Logisztikai menedzser");
62             createWorking(document, root, "6", "3", "6", "Logisztikai menedzser");

```

```

56     root.appendChild(document.createComment("Dolgozik"));
57     createWorking(document, root, "1", "1", "1", "Logisztikai menedzser");
58     createWorking(document, root, "2", "1", "2", "Logisztikai menedzser");
59     createWorking(document, root, "3", "2", "3", "Logisztikai menedzser");
60     createWorking(document, root, "4", "2", "4", "Logisztikai menedzser");
61     createWorking(document, root, "5", "3", "5", "Logisztikai menedzser");
62     createWorking(document, root, "6", "3", "6", "Logisztikai menedzser");
63
64     // Dolgozó
65     root.appendChild(document.createComment("Dolgozó"));
66     createWorker(document, root, "1", "Kovács Béla");
67     createWorker(document, root, "2", "Kiss János");
68     createWorker(document, root, "3", "Eged Csanád");
69     createWorker(document, root, "4", "Kun Tamás");
70     createWorker(document, root, "5", "Jámbor Andrea");
71     createWorker(document, root, "6", "Tóth Sándor");
72
73
74     // Vezeti
75     root.appendChild(document.createComment("Vezeti"));
76     createLeading(document, root, "1", "1", "1", "2017-01-01");
77     createLeading(document, root, "2", "2", "3", "2019-04-15");
78     createLeading(document, root, "3", "3", "6", "2020-09-07");
79
80
81     // Részleg
82     root.appendChild(document.createComment("Részleg"));
83     createDivision(document, root, "1", "Élelmiszer");
84     createDivision(document, root, "2", "Ital");
85     createDivision(document, root, "3", "Informatika");
86
87
88     // Documentum felépítése
89     TransformerFactory transformerFactory = TransformerFactory.newInstance();
90     Transformer transformer = transformerFactory.newTransformer();
91     transformer.setOutputProperty(OutputKeys.ENCODING, "UTF-8");
92     transformer.setOutputProperty(OutputKeys.INDENT, "yes");
93     transformer.setOutputProperty("{https://xml.apache.org/xslt}indent-amount", "2");
94
95     // Kimeneti fájl
96     DOMSource source = new DOMSource(document);
97     File outputFile = new File("XMLeio1rq1.xml");
98     StreamResult file = new StreamResult(outputFile);
99     transformer.transform(source, file);
100
101     // Konzolra kiírás
102     StreamResult console = new StreamResult(System.out);
103     transformer.transform(source, console);
104 }catch(Exception e) {
105     e.printStackTrace();

```

```

98         StreamResult file = new StreamResult(outputFile);
99         transformer.transform(source, file);
100
101         //Konzolra kiírás
102         StreamResult console = new StreamResult(System.out);
103         transformer.transform(source, console);
104     } catch (Exception e) {
105         e.printStackTrace();
106     }
107
108 }
109
110 public static void createOrder(Document document, Element root, String id, String div_fk, String name,
111     String amount, String delivered, String irsz, String city, String dest) {
112
113     Element order = document.createElement("rendelés");
114     order.setAttribute("RENDELÉS_ID", id);
115     order.setAttribute("RÉSZLEG_FK", div_fk);
116
117
118     Element _name = createElementValue(document, "név", name);
119     Element _amount = createElementValue(document, "mennyiség", amount);
120     Element _delivered = createElementValue(document, "leszállítva", delivered);
121
122     Element _destination = document.createElement("szállítási_hely");
123     Element _irsz = createElementValue(document, "irányítószám", irsz);
124     Element _city = createElementValue(document, "város", city);
125     Element _dest = createElementValue(document, "utca_hátszám", dest);
126
127     _destination.appendChild(_irsz);
128     _destination.appendChild(_city);
129     _destination.appendChild(_dest);
130
131     order.appendChild(_name);
132     order.appendChild(_amount);
133     order.appendChild(_delivered);
134     order.appendChild(_destination);
135
136     root.appendChild(order);
137 }
138
139 public static void createOrderConn(Document document, Element root, String id, String order_fk, String client_fk,
140     String date) {
141
142     Element orderConn = document.createElement("rendel");
143     orderConn.setAttribute("RENDEL_ID", id);
144     orderConn.setAttribute("RENDELÉS_FK", order_fk);
145     orderConn.setAttribute("MEGRENDELŐ_FK", client_fk);
146
147     Element _date = createElementValue(document, "rendelés_dátuma", date);
148     orderConn.appendChild(_date);

```

```

146
147     Element _date = createElementValue(document, "rendelés_dátuma", date);
148     orderConn.appendChild(_date);
149
150     root.appendChild(orderConn);
151 }
152
153 public static void createClient(Document document, Element root, String id, String name,
154     String phone, String email) {
155
156     Element orderClient = document.createElement("megrendelő");
157     orderClient.setAttribute("MEGRENDELŐ_ID", id);
158
159     Element _name = createElementValue(document, "cégnév", name);
160     Element _contact = document.createElement("elérhetőség");
161     Element _phone = createElementValue(document, "tel_száma", phone);
162     Element _email = createElementValue(document, "email", email);
163
164     _contact.appendChild(_phone);
165     _contact.appendChild(_email);
166
167     orderClient.appendChild(_name);
168     orderClient.appendChild(_contact);
169
170     root.appendChild(orderClient);
171 }
172
173 public static void createWorking(Document document, Element root, String id,
174     String div_fk, String worker_fk, String position) {
175
176     Element working = document.createElement("dolgozik");
177     working.setAttribute("DOLGOZIK_ID", id);
178     working.setAttribute("RÉSZLEG_FK", div_fk);
179     working.setAttribute("DOLGOZÓ_FK", worker_fk);
180
181     Element _position = createElementValue(document, "beosztás", position);
182     working.appendChild(_position);
183
184     root.appendChild(working);
185 }
186
187 public static void createWorker(Document document, Element root, String id, String name) {
188     Element worker = document.createElement("dolgozó");
189     worker.setAttribute("DOLGOZÓ_ID", id);
190
191     Element _name = createElementValue(document, "név", name);
192     worker.appendChild(_name);
193
194     root.appendChild(worker);

```

```

190
191     Element _name = createElementValue(document, "név", name);
192     worker.appendChild(_name);
193
194     root.appendChild(worker);
195 }
196
197 public static void createLeading(Document document, Element root, String id, String div_fk,
198     String worker_fk, String date) {
199     Element leading = document.createElement("vezeti");
200     leading.setAttribute("VEZETI_ID", id);
201     leading.setAttribute("RÉSZLEG_FK", div_fk);
202     leading.setAttribute("DOLGOZÓ_FK", worker_fk);
203
204     Element _date = createElementValue(document, "mikortól", date);
205     leading.appendChild(_date);
206
207     root.appendChild(leading);
208 }
209
210 public static void createDivision(Document document, Element root, String id, String name) {
211     Element division = document.createElement("részleg");
212     division.setAttribute("RÉSZLEG_ID", id);
213
214     Element _name = createElementValue(document, "név", name);
215     division.appendChild(_name);
216
217     root.appendChild(division);
218 }
219
220 public static Element createElementValue(Document doc, String name, String value) {
221     Element elem = doc.createElement(name);
222     elem.appendChild(doc.createTextNode(value));
223     return elem;
224 }
225 }
226

```