

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Autó kölcsönző rendszer

Készítette: **Markó Roland**

Neptunkód: **FRNJTJ**

Feladat leírása

A feladatom egy személyi autó kölcsönző rendszert modellez le.

A különböző elemek úgy látszódnak, ahogy egy külsős személy látná azt, ugyanis nem lehet pontosabb kényes részleteket látni, mint pl.: melyik személy milyen autót bérel, csak hogy azt teszi az adott cég(ek)nél, valamint azt sem lehet látni, hogy azért mennyit fizet. Ezért a kölcsönző felőli további belső információk nem látszódnak (pl.: a dolgozók adatai, a kölcsönző üzemeltetési ideje, stb...).

Az entitások, amik megjelennek a feladatomban a következők:

- Személy
- Kölcsönző
- Jármű
- Település

Ezek között az alábbi kapcsolatok szerepelnek:

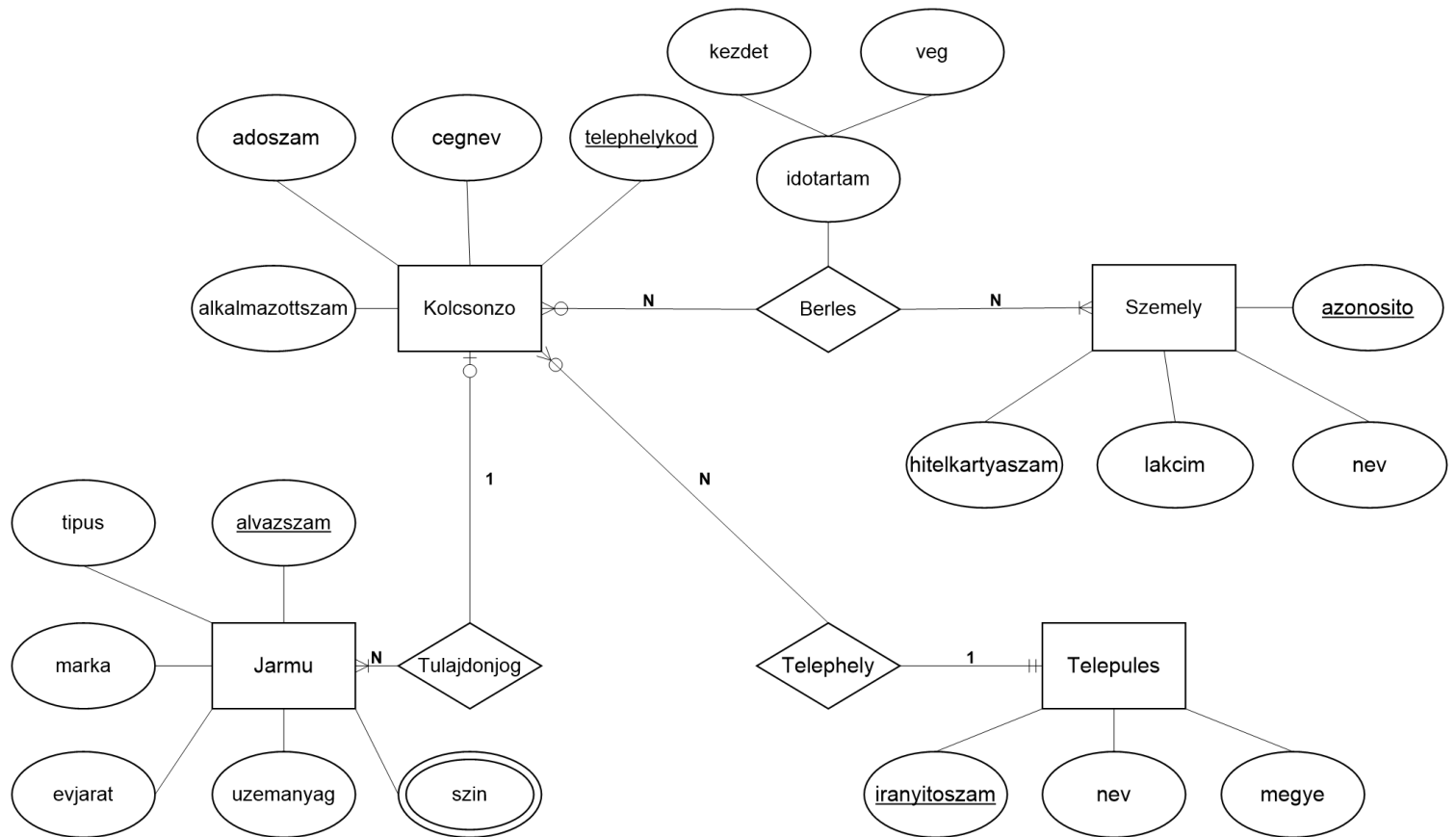
- Bérlés (Személy - Kölcsönző)
- Telephely (Kölcsönző - Település)
- Tulajdonjog (Kölcsönző - Jármű)

A későbbi jobb kialakítás és szervezés miatt a többértékű/többször megjelenő elemek egy összefoglaló elembe fognak kerülni.

A kölcsönző egy központi elem, ahol az összes többi elemmel (egy kapcsolat folyamán) összeköttetésben van.

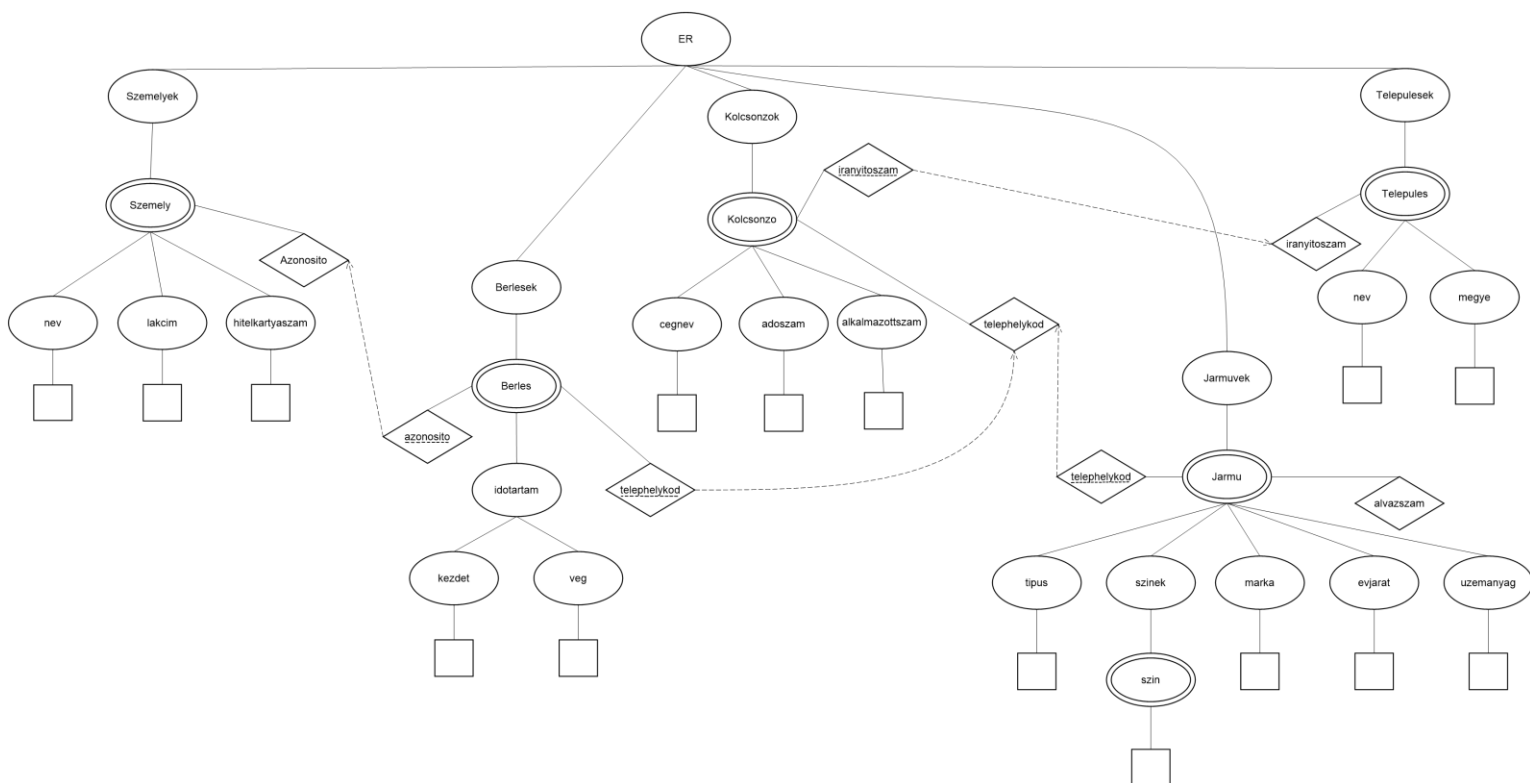
1. Feladat

1,a) Az adatbázis ER modell:



1b) Az adatbázis konvertálása XDM modellre:

ER-nek elnevezett gyöker elemből indul ki minden, és ezek közvetlen gyerekei az összefogó elemek, amik magába foglalják az akár többször előforduló elemeket (pl: Személy, Jármű, stb...). Már ez az ábra megjeleníti azokat az attribútumokat is, amik később idegen kulcsnak lesznek felhasználva.



1c) Az XDM modell alapján XML dokumentum készítése:

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

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

```
<Szemelyek>
```

```
<Szemely azonosito="951264TA">
```

```
<nev>Markó Roland</nev>
```

```
<lakcim>Bükkábrány, Arany János utca, 23</lakcim>
```

```
<hitelkartyaszam>12412323-61241235</hitelkartyaszam>
```

```
</Szemely>
<Szemely azonosito="312571DA">
  <nev>Lakatos Béla</nev>
  <lakcim>Sály, Fő út, 76</lakcim>
  <hitelkartyaszam>12318321-61543235</hitelkartyaszam>
</Szemely>
<Szemely azonosito="714232TA">
  <nev>Lengyel Bertalan</nev>
  <lakcim>Balogház, Kanizsa út, 64</lakcim>
  <hitelkartyaszam>19512323-62433135</hitelkartyaszam>
</Szemely>
<Szemely azonosito="614123TA">
  <nev>Szabó Lajos</nev>
  <lakcim>Épphely, Boldog utca, 56</lakcim>
  <hitelkartyaszam>73522323-62445325</hitelkartyaszam>
</Szemely>
<Szemely azonosito="612432TA">
  <nev>Hazadi Vendel</nev>
  <lakcim>Lator, Erdő út, 14</lakcim>
  <hitelkartyaszam>49152423-62444835</hitelkartyaszam>
</Szemely>
<Szemely azonosito="965414TA">
  <nev>Nándor Salamon</nev>
  <lakcim>Budapest, Szent Endre út, 97</lakcim>
  <hitelkartyaszam>13153323-63441335</hitelkartyaszam>
</Szemely>
</Szemelyek>

<Telepulesek>
<Telepules iranyitoszam="1000">
  <nev>Budapest</nev>
```

```
        <megye>Pest</megye>
    </Telepules>
    <Telepules irányitoszam="1001">
        <nev>Budapest</nev>
        <megye>Pest</megye>
    </Telepules>
    <Telepules irányitoszam="3422">
        <nev>Bükkábrány</nev>
        <megye>Borsod-Abaúj-Zemplén</megye>
    </Telepules>
    <Telepules irányitoszam="3400">
        <nev>Budapest</nev>
        <megye>Borsod-Abaúj-Zemplén</megye>
    </Telepules>
    <Telepules irányitoszam="3300">
        <nev>Eger</nev>
        <megye>Heves</megye>
    </Telepules>
    <Telepules irányitoszam="2574">
        <nev>Kiskunfogadó</nev>
        <megye>Szabolcs</megye>
    </Telepules>
</Telepulesek>

<Kolcsonzok>
    <Kolcsonzo telephelykod="1" irányitoszam="1001">
        <cegnev>Ubihard</cegnev>
        <adoszam>56125134321</adoszam>
        <alkalmazottszam>927</alkalmazottszam>
    </Kolcsonzo>
    <Kolcsonzo telephelykod="2" irányitoszam="3300">
```

```
<cegnev>Minecoin</cegnev>
<adoszam>54125134123</adoszam>
<alkalmazottszam>76</alkalmazottszam>
</Kolcsonzo>
<Kolcsonzo telephelykod="3" irányitoszam="3422">
  <cegnev>BitHuf</cegnev>
  <adoszam>85671279924</adoszam>
  <alkalmazottszam>412</alkalmazottszam>
</Kolcsonzo>
<Kolcsonzo telephelykod="4" irányitoszam="2574">
  <cegnev>Opel</cegnev>
  <adoszam>16802211838</adoszam>
  <alkalmazottszam>54</alkalmazottszam>
</Kolcsonzo>
<Kolcsonzo telephelykod="5" irányitoszam="3422">
  <cegnev>JSzF</cegnev>
  <adoszam>70746874303</adoszam>
  <alkalmazottszam>341</alkalmazottszam>
</Kolcsonzo>
<Kolcsonzo telephelykod="6" irányitoszam="1001">
  <cegnev>DollLove</cegnev>
  <adoszam>91394880987</adoszam>
  <alkalmazottszam>320</alkalmazottszam>
</Kolcsonzo>
<Kolcsonzo telephelykod="7" irányitoszam="1001">
  <cegnev>Minecoin</cegnev>
  <adoszam>54151908123</adoszam>
  <alkalmazottszam>5</alkalmazottszam>
</Kolcsonzo>
</Kolcsonzok>
```

<Jarmuvek>

<Jarmu alvazszam="512312331" telephelykod="2">

<tipus>Corsa</tipus>

<szinek>

<szin>Fekete</szin>

</szinek>

<marka>Opel</marka>

<evjarat>1998</evjarat>

<uzemanyag>Benzin</uzemanyag>

</Jarmu>

<Jarmu alvazszam="612312331" telephelykod="2">

<tipus>Golf</tipus>

<szinek>

<szin>Piros</szin>

</szinek>

<marka>Volkswagen</marka>

<evjarat>2010</evjarat>

<uzemanyag>Elektromos</uzemanyag>

</Jarmu>

<Jarmu alvazszam="639627878" telephelykod="3">

<tipus>Megan</tipus>

<szinek>

<szin>Szürke</szin>

</szinek>

<marka>Renault</marka>

<evjarat>1999</evjarat>

<uzemanyag>Benzin</uzemanyag>

</Jarmu>

<Jarmu alvazszam="383906114" telephelykod="2">

<tipus>Viper</tipus>

<szinek>


```
        <szin>Piros</szin>
        <szin>Narancs</szin>
        <szin>Sárga</szin>
    </szinek>
    <marka>Dodge</marka>
    <evjarat>2012</evjarat>
    <uzemanyag>Benzin</uzemanyag>
</Jarmu>
<Jarmu alvazszam="949498212" telephelykod="1">
    <tipus>Corsa</tipus>
    <szinek>
        <szin>Fekete</szin>
    </szinek>
    <marka>Opel</marka>
    <evjarat>2020</evjarat>
    <uzemanyag>Hibrid</uzemanyag>
</Jarmu>
<Jarmu alvazszam="834834873" telephelykod="6">
    <tipus>Touran</tipus>
    <szinek>
        <szin>Sötétkék</szin>
    </szinek>
    <marka>Volkswagen</marka>
    <evjarat>2004</evjarat>
    <uzemanyag>Dízel</uzemanyag>
</Jarmu>
</Jarmuvek>

<Berlesek>
    <Berles telephelykod="2" azonosito="714232TA">
        <idotartam>
```

```

        <kezdet>2000-10-12</kezdet>
        <veg>2000-10-13</veg>
    </idotartam>
</Berles>
<Berles telephelykod="6" azonosito="614123TA">
    <idotartam>
        <kezdet>2020-12-20</kezdet>
        <veg>2021-01-05</veg>
    </idotartam>
</Berles>
<Berles telephelykod="3" azonosito="312571DA">
    <idotartam>
        <kezdet>2021-10-15</kezdet>
    </idotartam>
</Berles>
<Berles telephelykod="2" azonosito="312571DA">
    <idotartam>
        <kezdet>2019-08-25</kezdet>
        <veg>2021-04-10</veg>
    </idotartam>
</Berles>
</Berlesek>
</ER>

```

1d) Az XML dokumentum alapján XMLSchema készítése (saját típusok):

```

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

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="ER">

```

```

<xs:complexType>
  <xs:sequence>
    <xs:element name="Szemelyek">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Szemely" maxOccurs="unbounded">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="nev"
type="xs:string"/>
                <xs:element name="lakcim"
type="xs:string"/>
                <xs:element name="hitelkartyaszam"
type="xs:string"/>
              </xs:sequence>
              <xs:attribute name="azonosito"
type="igazolvanyAzonosito" use="required"/>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="Telepulesek">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Telepules"
maxOccurs="unbounded">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="nev"
type="xs:string"/>
                <xs:element name="megye"
type="xs:string"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

```

                                <xs:attribute name="iranyitoszam"
type="iranyitosz" use="required"/>
                        </xs:complexType>
                </xs:element>
        </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Kolcsonzok">
        <xs:complexType>
                <xs:sequence>
                        <xs:element name="Kolcsonzo"
maxOccurs="unbounded">
                                <xs:complexType>
                                        <xs:sequence>
                                                <xs:element name="cegnev"
type="xs:string"/>
                                                <xs:element name="adoszam"
type="adoszam"/>
                                                <xs:element name="alkalmazottszam"
type="alkalmazottakSzama"/>
                                        </xs:sequence>
                                        <xs:attribute name="telephelykod"
type="xs:string" use="required"/>
                                        <xs:attribute name="iranyitoszam"
type="iranyitosz" use="required"/>
                                </xs:complexType>
                        </xs:element>
                </xs:sequence>
        </xs:complexType>
</xs:element>
<xs:element name="Jarmuvek">
        <xs:complexType>
                <xs:sequence>
                        <xs:element name="Jarmu" maxOccurs="unbounded">

```

```

        <xs:complexType>
            <xs:sequence>
                <xs:element name="tipus"
type="xs:string"/>

                <xs:element name="szinek">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="szin"
type="xs:string" maxOccurs="unbounded"/>

                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="marka"
type="xs:string"/>

                <xs:element name="evjarat"
type="xs:gYear"/>

                <xs:element name="uzemanyag "
type="uzemanyag"/>

            </xs:sequence>
            <xs:attribute name="alvazszam"
type="xs:string" use="required"/>

            <xs:attribute name="telephelykod"
type="xs:string" use="required"/>

        </xs:complexType>
    </xs:element>

</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Berlesek">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Berles" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>

```

```

        <xs:element name="idotartam">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="kezdet"
type="xs:date"/>
                    <xs:element name="veg"
type="xs:date" minOccurs="0"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
    <xs:attribute name="telephelykod"
type="xs:string" use="required"/>
    <xs:attribute name="azonosito"
type="xs:string" use="required"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>

<xs:key name="SzemelyK">
    <xs:selector xpath="Szemelyek/Szemely"/>
    <xs:field xpath="@azonosito"/>
</xs:key>

<xs:key name="TelepulesK">
    <xs:selector xpath="Telepulesek/Telepules"/>
    <xs:field xpath="@iranyitoszam"/>
</xs:key>

<xs:key name="KolcsonzoK">

```

```

        <xs:selector xpath="Kolcsonzok/Kolcsonzo"/>
        <xs:field xpath="@telephelykod"/>
    </xs:key>
    <xs:key name="JarmuK">
        <xs:selector xpath="Jarmuvek/Jarmu"/>
        <xs:field xpath="@alvazszam"/>
    </xs:key>

    <xs:keyref refer="SzemelyK" name="BerlesSzemely">
        <xs:selector xpath="Berlesek/Berles"/>
        <xs:field xpath="@azonosito"/>
    </xs:keyref>
    <xs:keyref refer="KolcsonzoK" name="BerlesKolcsonzo">
        <xs:selector xpath="Berlesek/Berles"/>
        <xs:field xpath="@telephelykod"/>
    </xs:keyref>
    <xs:keyref refer="TelepulesK" name="KolcsonzoFK">
        <xs:selector xpath="Kolcsonzok/Kolcsonzo"/>
        <xs:field xpath="@iranyitoszam"/>
    </xs:keyref>
    <xs:keyref refer="KolcsonzoK" name="JarmuFK">
        <xs:selector xpath="Jarmuvek/Jarmu"/>
        <xs:field xpath="@telephelykod"/>
    </xs:keyref>
</xs:element>

<xs:simpleType name="alkalmazottakSzama">
    <xs:restriction base="xs:int">
        <xs:minInclusive value="0"/>
    </xs:restriction>

```

```
</xs:simpleType>
<xs:simpleType name="iranyitosz">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="1000"/>
    <xs:maxExclusive value="5000"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="adoszam">
  <xs:restriction base="xs:string">
    <xs:minLength value="11"/>
    <xs:maxLength value="11"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="igazolvanyAzonosito">
  <xs:restriction base="xs:string">
    <xs:pattern value="([0-9]{6}[A-Z]A)"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="uzemanyag">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Benzin"/>
    <xs:enumeration value="Dízel"/>
    <xs:enumeration value="Elektromos"/>
    <xs:enumeration value="Hibrid"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>
```


2. Feladat

2a) adatolvasás (DOMReadFrnjtj.java):

Veszi az XML-ben levő elemeket és attribútumokat egy függvény rekurzív meghívásával, és ezeket egy StringBuilder típusú változóba menti. Órán mutatott példa szerint adja vissza az kiolvasott értéket.

```
package hu.domparse.frnjtj;

import java.io.IOException;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;

import org.w3c.dom.*;
import org.xml.sax.SAXException;

public class DOMReadFrnjtj {
    private final static String FILE_NAME = "XMLFRNJtJ.xml";
    private static StringBuilder text;
    private static int indent = 0;

    public static void main(String[] args) {
        Document doc;

        DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();

        text = new StringBuilder();           // text StringBuilder adattag példányosítása,
        később ebbe

        // kerülnek bele az értékek soronként

        try {
            DocumentBuilder db = dbf.newDocumentBuilder(); //Singleton példány lekérés
            doc = db.parse(FILE_NAME);                    //XML fájl-ra rácsatlakozás
        } catch (ParserConfigurationException | SAXException | IOException e) {
            e.printStackTrace();
        }
    }
}
```

```

Node er = doc.getFirstChild(); //ROOT elem lekérése

NodeList erList = er.getChildNodes(); //ROOT elem
tagjainak lekérése

getAllElements(erList); //
    Átadás egy függvénynek, ami

//    feldarabolja a dokumentumot
} catch (ParserConfigurationException | SAXException | IOException e) {
e.printStackTrace();
}

writeOut(); //text változó kiírása
}

private static void getAllElements(NodeList nl) { //    Az elem sorokat veszi és eldönti, hogy van-e
benne még több elem,

Node node; //    vagy
már csak szöveg érték

for (int i = 0; i < nl.getLength(); i++) {
node = nl.item(i);

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

text.append(indent(3)+node.getNodeName());

if(node.hasAttributes()) { //
    Amennyiben van attribútum,

text.append(" "+getAttributes(node.getAttributes()+"\n");//    a megfelelő taghoz hozzá adja
} else {
text.append("\n");
}

if(node.getChildNodes().getLength() != 1) { //Ha a hossza nem egy, akkor nem csak szöveg
tartalma van

```

```

NodeList childList = node.getChildNodes();

indent++;

getAllElements(childList);

} else {                                     //Ha van értéke adja hozzá a text-hez

indent++;
text.append(indent(3)+node.getTextContent()+"\n");
indent--;

}
}
}
indent--;
}

private static String getAttributes(NamedNodeMap node) {    //      kiírja az attribútumokat {tag :
érték; tag : érték}

StringBuilder sb = new StringBuilder("{}");                //      formában

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

if(i != node.getLength()-1) {

sb.append(node.item(i).getNodeName()+" : "+node.item(i).getNodeValue()+" ; ");

} else {

sb.append(node.item(i).getNodeName()+" : "+node.item(i).getNodeValue());

}

}

sb.append("{}");

return sb.toString();

}

```

```
private static String indent(int multiplyBy) {
    bentebb törését
```

```
//Függvény ami számontartja a bekezdés
```

```
StringBuilder whiteSpaces = new StringBuilder();
```

```
for (int i = 0; i < (indent*multiplyBy); i++) {
```

```
    whiteSpaces.append(" ");
```

```
}
```

```
return whiteSpaces.toString();
```

```
}
```

```
private static void writeOut() {
```

```
    System.out.println(text.toString());
```

```
}
```

```
}
```

Szemelyek	Borsod-Abaúj-Zemplén	tipus
Szemely {azonosito : 951264TA}	Telepules {iranyitoszam : 3300}	Corsa
nev	nev	szinek
Markó Roland	Eger	szin
lakcim	megye	Fekete
Bükkábrány, Arany János utca, 23	Heves	marka
hitelkartyaszam	Telepules {iranyitoszam : 2574}	Opel
12412323-61241235	nev	evjarat
Szemely {azonosito : 312571DA}	Kiskunföld	1998
nev	megye	uzemanyag
Lakatos Béla	Szabolcs	Benzin
lakcim	Kolcsonzo	Jarmu {alvazszam : 612312331; telephelykod : 2}
Sály, Fő út, 76	Kolcsonzo {iranyitoszam : 1001; telephelykod : 1}	tipus
hitelkartyaszam	cegnev	Golf
12318321-61543235	Ubihard	szinek
Szemely {azonosito : 714232TA}	adoszam	szin
nev	56125134321	Piros
Lengyel Bertalan	alkalmazottszam	marka
lakcim	927	Volkswagen
Balogház, Kanizsa út, 64	Kolcsonzo {iranyitoszam : 3300; telephelykod : 2}	evjarat
hitelkartyaszam	cegnev	2010
19512323-624433135	Mincoin	uzemanyag
Szemely {azonosito : 614123TA}	adoszam	Elektromos
nev	54125134123	Jarmu {alvazszam : 639627878; telephelykod : 3}
Szabó Lajos	alkalmazottszam	tipus
lakcim	76	Megan
Epphely, Boldog utca, 56	Kolcsonzo {iranyitoszam : 3422; telephelykod : 3}	szinek
hitelkartyaszam	cegnev	szin
73522323-62445325	BitHuf	Szürke
Szemely {azonosito : 612432TA}	adoszam	marka
nev	85671279924	Renault
Hazadi Vendel	alkalmazottszam	evjarat
lakcim	412	1999
Lator, Erdő út, 14	Kolcsonzo {iranyitoszam : 2574; telephelykod : 4}	uzemanyag
hitelkartyaszam	cegnev	Benzin
49152423-62444835	Opel	Jarmu {alvazszam : 383906114; telephelykod : 2}
Szemely {azonosito : 965414TA}	adoszam	tipus
nev	16802211838	Viper
Nándor Salamon	alkalmazottszam	szinek
lakcim	54	szin
Budapest, Szent Endre út, 97	Kolcsonzo {iranyitoszam : 3422; telephelykod : 5}	Piros
hitelkartyaszam	cegnev	szin
13153323-63441335	JSZF	Narancs
Telepulesek	adoszam	szin
Telepules {iranyitoszam : 1000}	70746874303	Sárga
nev	alkalmazottszam	marka
Budapest	341	Dodge
megye	Kolcsonzo {iranyitoszam : 1001; telephelykod : 6}	evjarat
Pest	cegnev	2012
Telepules {iranyitoszam : 1001}	Dolllove	uzemanyag
nev	adoszam	Benzin
Budapest	91394880987	Jarmu {alvazszam : 949498212; telephelykod : 1}
megye	alkalmazottszam	tipus
Pest	320	Corsa
Telepules {iranyitoszam : 3422}	Kolcsonzo {iranyitoszam : 1001; telephelykod : 7}	szinek
nev	cegnev	szin
Bükkábrány	Mincoin	Fekete
megye	adoszam	marka
Borsod-Abaúj-Zemplén	54151908123	Opel
Telepules {iranyitoszam : 3400}	alkalmazottszam	evjarat
nev	5	2020
Budapest	Jarmuvek	uzemanyag
megye	Jarmu {alvazszam : 512312331; telephelykod : 2}	Hibrid

2b) adatmódosítás (DOMModifyFrnjtj.java):

Módosítás történhet elem és attribútum szerint, ez a kettő akár történhet egyszerre a lefutás során. A kiírás transformáció segítségével történik. Ha a megváltoztató attribútum, vagy elem nevét nem adjuk meg, akkor abból nem történik módosítás.

```
package hu.domparse.frnjtj;
```

```
import java.io.IOException;
```

```
import javax.xml.parsers.*;
```

```
import javax.xml.transform.*;
```

```
import javax.xml.transform.dom.DOMSource;
```

```
import javax.xml.transform.stream.StreamResult;
```

```
import org.w3c.dom.*;
```

```
import org.xml.sax.SAXException;
```

```
public class DOMModifyFrnjtj {
```

```
    private final static String FILE_NAME = "XMLFRNJtj.xml";
```

```
    private final static String ROOT_OF_ELEMENT = "Szemelyek",           //A
    módosítandó elem ROOT-ja
```

```
    ELEMENT_TO_MODIFY = "nev",                                         //A módosítandó mező neve, ha
    üres, nem történik változtatás
```

```
    MODIFIABLE_VALUE_OF_ELEMENT = "Markó Roland", //Mező beli érték, amit módosítani
    szeretnénk, ha üres, minden értéket változtat
```

```
    MODIFY_VALUE_OF_ELEMENT_TO = "Molnar Mark"; //Érték amire változtatja a módosítandó
    mezőt
```

```
    private final static String ROOT_OF_ATTRIBUTE = "Kolcsonzok",      //A módosítandó
    attribútum ROOT-ja
```

```
    ATTRIBUTE_TO_MODIFY = "iranyitoszam",                             //A módosítandó attribútum neve, ha üres,
    nem történik változtatás
```

```
MODIFIABLE_VALUE_OF_ATTRIBUTE = "1001",           //attribútum beli érték, amit módosítani
szeretnénk, ha üres, minden értéket változtat

MODIFY_VALUE_OF_ATTRIBUTE_TO = "1003";           //Érték amire változtatja a
módosítandó attribútumot
```

```
public static void main(String[] args) {

    Document doc;

    DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();

    try {

        DocumentBuilder db = dbf.newDocumentBuilder();

        doc = db.parse(FILE_NAME);

        Node rootNode = doc.getFirstChild();

        NodeList rootNodeList = rootNode.getChildNodes();

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

            Node supNode = rootNodeList.item(i);

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

                //ELEMENT

                if(supNode.getNodeName().equals(ROOT_OF_ELEMENT)) {

                    if(!ELEMENT_TO_MODIFY.isBlank()) {

                        setElementValue(supNode.getChildNodes());

                    }

                }

                //ATTRIBUTE

                if(supNode.getNodeName().equals(ROOT_OF_ATTRIBUTE)) {

                    if(!ATTRIBUTE_TO_MODIFY.isBlank()) {

                        setAttributeValue(supNode.getChildNodes());

                    }

                }

            }

        }

    } catch (Exception e) {

        e.printStackTrace();

    }

}
```

```
}  
}  
}  
}
```

```
Transformer tf = TransformerFactory.newInstance().newTransformer();
```

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

```
tf.setOutputProperty(OutputKeys.INDENT, "yes");
```

```
tf.setOutputProperty("{http://xml.apache.org/xslt}indent-amount", "2");
```

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

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

```
System.out.println("----- Modified File -----");
```

```
tf.transform(source, console);
```

```
} catch (ParserConfigurationException | SAXException | IOException
```

```
| TransformerException | TransformerFactoryConfigurationError e) {
```

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

```
}
```

```
}
```

```
public static void setElementValue(NodeList nodeList) {           //Megkeresi a feltételeknek  
    megfelelő elemeket és a megadott értékre írja át azokat
```

```
    boolean found = false;
```

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

```
        Node node = nodeList.item(i);
```

```

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

    NodeList list = node.getChildNodes();

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

        Node subNode = list.item(j);

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

            if(subNode.getNodeName().equals(ELEMENT_TO_MODIFY)) { //vizsgálja, hogy a megfelelő elemnél
                van-e a ciklus

                if(subNode.getTextContent().equals(MODIFIABLE_VALUE_OF_ELEMENT) || //megnézi, hogy a
                    módosítandó érték megtalálható-e itt

                MODIFIABLE_VALUE_OF_ELEMENT.isBlank()) {

                    subNode.setTextContent(MODIFY_VALUE_OF_ELEMENT_TO);

                    found = true;

                }

            }

        }

    }

    if(!found)

        System.out.println("Nem történt elem változtatás");

}

public static void setAttributeValue(NodeList nodeList) { //Megkeresi a feltételeknek
    megfelelő attribútumokat és a megadott értékre írja át azokat

    boolean found = false;

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

```



```
Node node = nodeList.item(i);
```

```
if(node.getNodeType() == Node.ELEMENT_NODE) {    //Kiválszja a listából az elemeket
```

```
Element element = (Element) node;
```

```
if(!element.getAttribute(ATTRIBUTE_TO_MODIFY).isBlank()) {    //Megnézi, hogy létezik-e az a tulajdonság,
```

```
//amit meg akarunk változtatni
```

```
if(element.getAttribute(ATTRIBUTE_TO_MODIFY).equals(MODIFIABLE_VALUE_OF_ATTRIBUTE) ||  
MODIFIABLE_VALUE_OF_ATTRIBUTE.isBlank()) {
```

```
element.setAttribute(ATTRIBUTE_TO_MODIFY, MODIFY_VALUE_OF_ATTRIBUTE_TO);    //A megadott attribútum értékét a megadott értékre írja át
```

```
found = true;
```

```
}
```

```
}
```

```
}
```

```
}
```

```
if(!found)
```

```
System.out.println("Nem történt attribútum változtatás");
```

```
}
```

```
}
```

```
----- Modified File -----
<?xml version="1.0" encoding="UTF-8"?><ER xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="
XMLSchemaFRNJTD.xsd">
```

```
<Szemelyek>

<Szemely azonosito="951264TA">

  <nev>Molnar Mark</nev>

  <lakcim>B|kk|ibn|iny, Arany J|inos utca, 23</lakcim>

  <hitelkartyaszam>12412323-61241235</hitelkartyaszam>

</Szemely>

<Szemely azonosito="312571DA">

  <nev>Lakatos B|ela</nev>

  <lakcim>S|ily, F|l|t, 76</lakcim>

  <hitelkartyaszam>12318321-61543235</hitelkartyaszam>

</Szemely>

<Szemely azonosito="714232TA">

  <nev>Lengyel Bertalan</nev>

  <lakcim>Balogh|iz, Kanizsa |t, 64</lakcim>

  <hitelkartyaszam>19512323-62433135</hitelkartyaszam>

</Szemely>

<Szemely azonosito="614123TA">

  <nev>Szab|l Lajos</nev>

  <lakcim>|epphely, Boldog utca, 56</lakcim>

  <hitelkartyaszam>73522323-62445325</hitelkartyaszam>

</Szemely>

<Szemely azonosito="612432TA">

  <nev>Hazadi Vendel</nev>

  <lakcim>Lator, Erd|l|t, 14</lakcim>

  <hitelkartyaszam>49152423-62444835</hitelkartyaszam>

</Szemely>

<Szemely azonosito="965414TA">

  <nev>N|indor Salamon</nev>

  <lakcim>Budapest, Szent Endre |t, 97</lakcim>

  <hitelkartyaszam>13153323-63441335</hitelkartyaszam>
```

```
<Kolcsonzo iranyitoszam="2574" telephelykod="4">

  <cegnev>Opel</cegnev>

  <adoszam>16802211838</adoszam>

  <alkalmazottsam>54</alkalmazottsam>

</Kolcsonzo>

<Kolcsonzo iranyitoszam="3422" telephelykod="5">

  <cegnev>JSzf</cegnev>

  <adoszam>70746874303</adoszam>

  <alkalmazottsam>341</alkalmazottsam>

</Kolcsonzo>

<Kolcsonzo iranyitoszam="1003" telephelykod="6">

  <cegnev>DollLove</cegnev>

  <adoszam>91394880987</adoszam>

  <alkalmazottsam>320</alkalmazottsam>

</Kolcsonzo>

<Kolcsonzo iranyitoszam="1003" telephelykod="7">

  <cegnev>Minecoin</cegnev>

  <adoszam>54151908123</adoszam>

  <alkalmazottsam>5</alkalmazottsam>

</Kolcsonzo>

</Kolcsonzok>
```

3c) adatlekérdezés (DOMQueryFrnjtj.java):

A program elején konstansként megadhatjuk a lekérdezendő Node elemet, annak attribútumát, és értékét, és ezek alapján nézi át, adja át a sorokat egy StringBuilder típusú változónak majd írhatja ki az adatokat.

```
package hu.domparsed.frnjtj;

import java.io.IOException;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;

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

public class DOMQueryFrnjtj {
    private final static String FILE_NAME = "XMLFRNJtJ.xml";
    private final static String QUERY_BY_NODE = "Kölcsönzok";
    private final static String QUERY_BY_ATTRIBUTE = "iranyitoszam";
    private final static String QUERY_BY_ATTRIBUTEVALUE = "1001";
    private static StringBuilder text;

    public static void main(String[] args) {
        Document doc;

        DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();

        text = new StringBuilder("Query from:
"+QUERY_BY_NODE+"\n_____ \n\n");
```

```

try {
    DocumentBuilder db = dbf.newDocumentBuilder();
    doc = db.parse(FILE_NAME);

    NodeList erList = doc.getFirstChild().getChildNodes();    //ROOT-ba levő elemek

    for (int i = 0; i < erList.getLength(); i++) {            //Kiszedi az 5 fő tagot
        Node node = erList.item(i);

        if(node.getNodeType() == Node.ELEMENT_NODE &&
            node.getNodeName().equals(QUERY_BY_NODE)) {

            NodeList subList = node.getChildNodes();          //Többször előforduló elemek
            QueryNodes(subList, QUERY_BY_ATTRIBUTE, QUERY_BY_ATTRIBUTEVALUE);
        }
    }

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

    writeOut();
}

public static void QueryNodes(NodeList nl, String attributeName, String attributeValue) {
    //megkeresi azokat a tagokat, amiknek az attribútumai megfelelnek a paraméterben átadott
    értékekkel

    for (int i = 0; i < nl.getLength(); i++) {    //Többször előforduló elemek
        Node node = nl.item(i);

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

```

```
String[] attributes = getAttributes(node.getAttributes()).split(";");
```

```
String[] firstAttribute = attributes[0].split(":");
```

```
if(attributes.length != 1) {
```

```
String[] secondAttribute = attributes[1].split(":");
```

```
if(secondAttribute[0].equals(attributeName) && secondAttribute[1].equals(attributeValue)) {
```

```
addNodeToText(node, attributes);
```

```
continue;
```

```
}
```

```
}
```

```
if(firstAttribute[0].equals(attributeName) && firstAttribute[1].equals(attributeValue)) {
```

```
addNodeToText(node, attributes);
```

```
}
```

```
}
```

```
}
```

```
}
```

```
private static void addNodeToText(Node node, String[] attributes) {    //a text adattaghoz újabb  
szöveg táblát ad hozzá
```

```
text.append("Current Element: "+node.getNodeName()+"\n");
```

```
text.append("Attributes: \n\t{"+attributes[0]);
```

```
if(attributes.length != 1) {
```

```
text.append(" ; "+attributes[1]+"}\n");
```

```
} else {
```

```
text.append("}\n");
```

```
}
```

```
text.append("Elements: \n"+getElements(node)+"\n");
```

```
text.append("-----\n");
```

```
}
```

```
private static String getAttributes(NamedNodeMap node) {    //visszaadja az attribútumokat  
tag:érték;tag:érték alakba
```

```
StringBuilder sb = new StringBuilder();
```

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

```
if(i != node.getLength()-1) {
```

```
sb.append(node.item(i).getNodeName()+":"+node.item(i).getNodeValue()+";");
```

```
} else {
```

```
sb.append(node.item(i).getNodeName()+":"+node.item(i).getNodeValue());
```

```
}
```

```
}
```

```
return sb.toString();
```

```
}
```

```
private static String getElements(Node node) { //az elemeket egy olvasható szöveg alakba hozza
```

```
StringBuilder sb = new StringBuilder();
```

```
NodeList list = node.getChildNodes();
```

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

```
Node subNode = list.item(i);
```

```
if(subNode.getNodeType() == Node.ELEMENT_NODE) {
```

```
sb.append("\t"+subNode.getNodeName()+": "+subNode.getTextContent()+"\n");
```

```
}
```

```
}
```

```
return sb.toString();
```

```
}
```

```
private static void writeOut() {
```

```
System.out.println(text.toString());
```

```
}
```

```
}
```

```
Query from: Kolcsonzok
```

```
-----  
Current Element: Kolcsonzo
```

```
Attributes:
```

```
{iranyitoszam:1001 ; telephelykod:1}
```

```
Elements:
```

```
cegnev: Ubihard
```

```
adoszam: 56125134321
```

```
alkalmazottszam: 927
```

```
-----  
Current Element: Kolcsonzo
```

```
Attributes:
```

```
{iranyitoszam:1001 ; telephelykod:6}
```

```
Elements:
```

```
cegnev: Dolllove
```

```
adoszam: 91394880987
```

```
alkalmazottszam: 320
```

```
-----  
Current Element: Kolcsonzo
```

```
Attributes:
```

```
{iranyitoszam:1001 ; telephelykod:7}
```

```
Elements:
```

```
cegnev: Minecoin
```

```
adoszam: 54151908123
```

```
alkalmazottszam: 5  
-----
```