

Jegyzőkönyv

Adatkezelés XML környezetben

Féléves feladat

Gyárak tasaknyomtatásának nyilvántartása

Készítette: László Andrea

Neptun kód: DJ7PNE

Gyakorlat: Szerda 10-12

Gyakorlatvezető: Dr. Bednarik László

A feladat témája:

A beadandó témája egy olyan adatbázis, amely több gyárnak a rendeléseit számon tudja tartani. Van lehetőség arra, hogy megnézzük az adott gyárak gépeit és azokhoz tartozó rendeléseket, illetve azokhoz tartozó vevőket.

- *Nyomtatógép egyed tulajdonságai*
 - Termék neve: A nyomtatógép egyedi kulcsa
 - Típus: A nyomtatógép típusa
 - Folyóméter: Az adott termékből nyomtatógép által nyomtatott mennyiség
 - Raszterhenger: Milyen fajta hengert használtak egyes színekhez
 - Festék felhasználása: Adott festék folyóméterenkénti felhasználása
- *Festék egyed tulajdonságai*
 - Pantonszám: A festék egyedi tulajdonsága
 - Gyártó: Adott festék gyártói
 - Raktáron lévő mennyiség: Éppen a gyárnál mekkora mennyiség van adott festékekből
 - Lejárat dátum: A festék lejáratainak dátuma
- *Tasakrendelés egyed tulajdonságai*
 - tasakID: A tasak egyedi kulcsa
 - Mennyiség: Mekkora mennyiséggel rendeltek (kg)
 - Termék neve: Adott termék neve, amelyet éppen rendelnek
 - Papír minősége: Milyen fajta papírból fogják gyártani az adott tasakot
- *Vevő egyed tulajdonságai*
 - Adószám: A vevő egyed kulcsa
 - Név: Vevő cég/személy neve
 - Telephely: Összetett tulajdonság, mely az irányítószámból és utcából áll
 - Származás: A vevő származási helye
- *Gyár egyed tulajdonságai*
 - Adószám: A gyár egyed kulcsa
 - Név: A gyár neve

- Telephely: A gyár telephelye, ahol elhelyezkedik
- Alapítási dátum: A gyár alapításának a dátuma
- *Logo egyed tulajdonságai*
 - Tervező: A logo tervezője
 - Színek: A logóban felhasznált színek
 - logoID: A verziószám, amely el lett fogadva az adott logóból
 - Elkészítés dátuma: A logo elkészítésének dátuma
- *Tulajdonos egyed tulajdonságai*
 - Tajsám: A tulajdonos egyedi kulcsa
 - Név: A tulajdonos neve
 - Születési dátum: Tulajdonos születési dátuma
 - Szerződés kezdete: Az a dátum, amikor tulajdonossá vált

Az egyedek közötti kapcsolatok:

Nyomtatógép, Tasakrendelés és Vevő:

Hármas kapcsolat van közöttük, a közös kapcsolatuk a szerződésszám.

Nyomtatógép és Festék:

Több több kapcsolat van közöttük, mert egy nyomtatógép több festéket is felhasználhat, és ugyanazt a festéket több nyomtatógép is felhasználhat.

Vevő és Gyár:

Több több kapcsolat van köztük, mert egy gyárnak lehet több vevője, de egy vevő több gyártól is rendelhet.

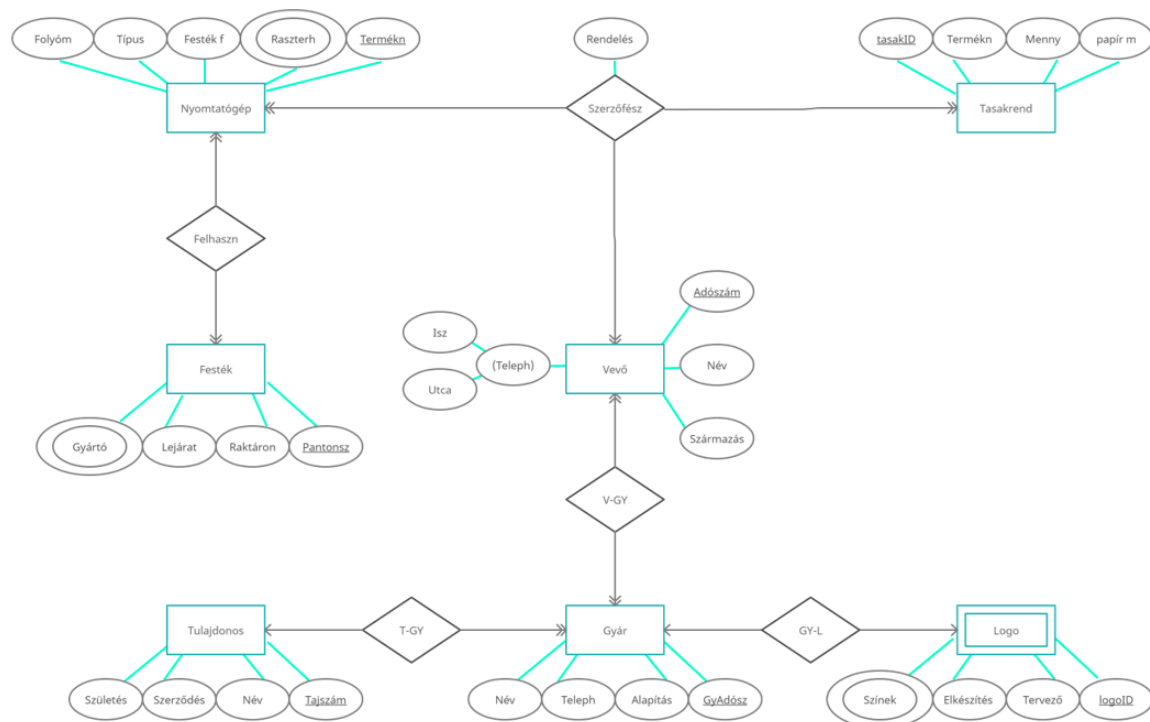
Gyár és Tulajdonos:

Egy több kapcsolat van közöttük, mivel egy tulajdonosnak lehet több gyára, viszont egy gyárnak egy fő tulajdonosa van.

Gyár és Logo:

Egy egy kapcsolat van közöttük, mivel egy gyárnak egy logója lehet, és egy logóhoz is egy gyár tartozhat.

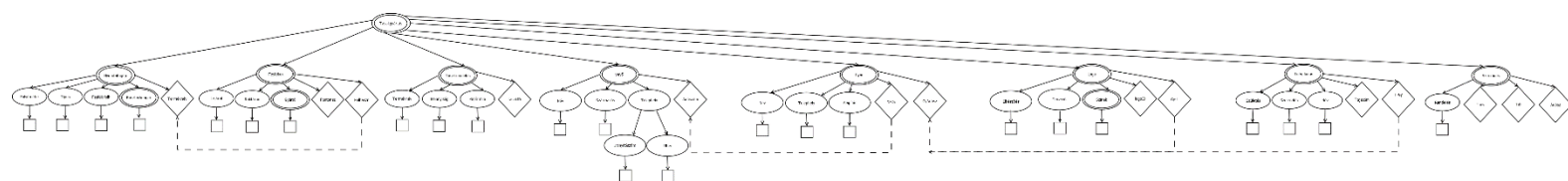
A feladat ER modellje:



Az ER modell konvertálása XDM modellé

XDM modellnél háromféle jelölés alkalmazunk. Ezek az ellipszis, a rombusz, illetve a téglalap. Az ellipszis jelöli az elemeket, minden egyedből elem lesz, ezen felül a tulajdonságokból is. A rombusz jelöli az attribútumokat, amelyek a kulcs tulajdonságokból keletkeznek. A téglalap jelöli a szöveget, amely majd az XML dokumentumban fog megjelenni. Azoknak az elemeknek, amelyek többször is előfordulhatnak, a jelölése dupla ellipszissel történik. Az idegenkulcsok és a kulcsok közötti kapcsolatot szaggatott vonalas nyílal jelöljük.

A feladat XDM modellje:



XML dokumentum készítése

Az XDM modell alapján az XML dokumentumot úgy készítettem el, hogy először is a root elementtel kezdtem, ami a tasakgyártás volt. Mindezek után a gyerekelemeiből legalább 3-3 példányt létrehoztam. Ezen elemeknek az attribútumai közé tartoznak a kulcsok, illetve idegenkulcsok is, mindezek után, pedig ezeknek a parent elementeknek is létrehoztam az elemeit. A többértékű elemeknél több elemet is létrehoztam, illetve volt olyan eset is, amikor ezeknek az elemeknek is voltak gyerekelemei.

XML dokumentum forráskódja:

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

```
<tasakgyartas xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
xs:noNamespaceSchemaLocation="XMLSchemaDJ7PNE.xsd">
```

```
<!--Nyomtatógépek-->
```

```
<nyomtatogep termeknev="1kg Gyermelyi finomliszt">
  <folyometer>2500</folyometer>
  <tipus>VF001</tipus>
  <festekfelhasznalas>5kg</festekfelhasznalas>
  <raszterhenger>180/10</raszterhenger>
  <raszterhenger>120/8</raszterhenger>
</nyomtatogep>
```

```
<nyomtatogep termeknev="2kg Hajdú búzadara">
  <folyometer>12000</folyometer>
  <tipus>VF001</tipus>
  <festekfelhasznalas>8kg</festekfelhasznalas>
  <raszterhenger>180/10</raszterhenger>
</nyomtatogep>
```

```
<nyomtatogep termeknev="5kg Auschan Optimum">
  <folyometer>20000</folyometer>
  <tipus>VF002</tipus>
  <festekfelhasznalas>14kg</festekfelhasznalas>
  <raszterhenger>120/8</raszterhenger>
  <raszterhenger>110/2</raszterhenger>
</nyomtatogep>
```

```
<!--Festékek-->
```

```
<festek pantonszam="P1495" felhasznalas="1kg Gyermelyi finomliszt">
  <lejarat>2030.10.15</lejarat>
  <raktaron>150kg</raktaron>
  <gyarto>Colorprint</gyarto>
</festek>
```

```
<festek pantonszam="P706" felhasznalas="1kg Gyermelyi finomliszt">
  <lejarat>2032.08.01</lejarat>
  <raktaron>450kg</raktaron>
  <gyarto>Colorprint</gyarto>
  <gyarto>Poli-Farbe</gyarto>
</festek>
```

```
<festek pantonszam="P1400" felhasznalas="2kg Hajdú búzadara">
  <lejarat>2028.02.22</lejarat>
```

<raktaron>150kg</raktaron>
<gyarto>Colorprint</gyarto>
</festek>

<festek pantonszam="P632" felhasznalas="5kg Auschan Optimum">
<lejarat>2024.07.19</lejarat>
<raktaron>100kg</raktaron>
<gyarto>Colorprint</gyarto>
</festek>

<festek pantonszam="P935" felhasznalas="5kg Auschan Optimum">
<lejarat>2026.12.31</lejarat>
<raktaron>50kg</raktaron>
<gyarto>Colorprint</gyarto>
</festek>

<festek pantonszam="P1250" felhasznalas="5kg Auschan Optimum">
<lejarat>2032.12.15</lejarat>
<raktaron>80kg</raktaron>
<gyarto>Colorprint</gyarto>
</festek>

<!--Tasakrendelések-->

<tasakrendeles tasakID="1">
<termeknev>1kg Gyermelyi finomliszt</termeknev>
<mennyiseg>150000db</mennyiseg>
<papirmin>80gr</papirmin>
</tasakrendeles>

<tasakrendeles tasakID="2">
<termeknev>2kg Hajdú búzadara</termeknev>
<mennyiseg>100000db</mennyiseg>
<papirmin>70gr</papirmin>
</tasakrendeles>

<tasakrendeles tasakID="3">
<termeknev>5kg Auschan Optimum</termeknev>
<mennyiseg>50000db</mennyiseg>
<papirmin>100gr</papirmin>
</tasakrendeles>

<!--Vevők-->

<vevo adoszam="77777777-77-777777">
<nev>Egyes cég</nev>
<szarmazas>Magyarország</szarmazas>
<telephely>
<isz>4090</isz>
<utca>Kossuth</utca>

</telephely>
<telephely>
 <isz>4090</isz>
 <utca>Bethlen</utca>
</telephely>
</vevo>

<vevo adoszam="22222222-22-222222">
 <nev>Kettes cég</nev>
 <szarmazas>Magyarország</szarmazas>
 <telephely>
 <isz>3515</isz>
 <utca>Diósgyőri</utca>
 </telephely>
</vevo>

<vevo adoszam="88888888-22-787878">
 <nev>Hármas cég</nev>
 <szarmazas>Magyarország</szarmazas>
 <telephely>
 <isz>2080</isz>
 <utca>Petőfi</utca>
 </telephely>
</vevo>

<!--Gyárak-->

<gyar gyAdoszam="12345678-22-654321" vGy="77777777-77-777777">
 <nev>Egyes gyár</nev>
 <telephely>Felsősima</telephely>
 <alapitas>2002</alapitas>
</gyar>

<gyar gyAdoszam="13151614-32-956348" vGy="77777777-77-777777">
 <nev>Kettes gyár</nev>
 <telephely>Miskolc</telephely>
 <alapitas>2008</alapitas>
</gyar>

<gyar gyAdoszam="32945817-68-164872" vGy="22222222-22-222222">
 <nev>Hármas gyár</nev>
 <telephely>Debrecen</telephely>
 <alapitas>2016</alapitas>
</gyar>

<gyar gyAdoszam="62599875-77-121212" vGy="88888888-22-787878">
 <nev>Négyes gyár</nev>
 <telephely>Debrecen</telephely>
 <alapitas>2018</alapitas>
</gyar>

<!--Logók-->

<logo logoId="1" gyL="12345678-22-654321">
 <elkeszites>2001.12.15</elkeszites>
 <szinek>Piros</szinek>
 <szinek>Kék</szinek>
 <szinek>Zöld</szinek>
 <tervezo>Kiss Péter</tervezo>
</logo>

<logo logoId="2" gyL="13151614-32-956348">
 <elkeszites>2007.12.20</elkeszites>
 <szinek>Fehér</szinek>
 <szinek>Zöld</szinek>
 <tervezo>Nagy Zoltán</tervezo>
</logo>

<logo logoId="3" gyL="32945817-68-164872">
 <elkeszites>2016.02.08</elkeszites>
 <szinek>Vörös</szinek>
 <szinek>Fehér</szinek>
 <szinek>Fekete</szinek>
 <tervezo>Bogyó Annamária</tervezo>
</logo>

<logo logoId="4" gyL="62599875-77-121212">
 <elkeszites>2018.06.26</elkeszites>
 <szinek>Fehér</szinek>
 <szinek>Fekete</szinek>
 <tervezo>Varga Krisztina</tervezo>
</logo>

<!--Tulajdonosok-->

<tulajdonos tajszam="119075100" tGy="12345678-22-654321">
 <nev>Terplán Zénó</nev>
 <szuletes>1982</szuletes>
 <szerezese>2002</szerezese>
</tulajdonos>

<tulajdonos tajszam="117080205" tGy="13151614-32-956348">
 <nev>Tóth László</nev>
 <szuletes>1970</szuletes>
 <szerezese>1993</szerezese>
</tulajdonos>

<tulajdonos tajszam="120154203" tGy="32945817-68-164872">
 <nev>Reszegi Ivett</nev>
 <szuletes>1969</szuletes>


```

        <szerezese>2000</szerezese>
    </tulajdonos>

    <tulajdonos tajsam="120005203" tGy="62599875-77-121212">
        <nev>Balogh Panna</nev>
        <szuletes>1993</szuletes>
        <szerezese>2012</szerezese>
    </tulajdonos>

    <!--Szerződés kapcsolat-->

    <szerezodes tNev="1kg Gyermei finomliszt" tasakID="1" adoszam="77777777-77-777777">
        <rendeles>2022.09.15</rendeles>
    </szerezodes>

    <szerezodes tNev="2kg Hajdú búzadara" tasakID="2" adoszam="77777777-77-777777">
        <rendeles>2022.09.18</rendeles>
    </szerezodes>

    <szerezodes tNev="5kg Auschan Optimum" tasakID="3" adoszam="88888888-22-787878">
        <rendeles>2022.10.06</rendeles>
    </szerezodes>

    <szerezodes tNev="5kg Auschan Optimum" tasakID="3" adoszam="22222222-22-222222">
        <rendeles>2022.10.03</rendeles>
    </szerezodes>

</tasakgyartas>

```

XMLSchema készítése

Az XML dokumentumhoz kellett a validációt elősegítő sémát létrehozni. Ehhez először létrehoztam az egyszerű elemeket, amelyekre a felépítésben referálni tudok, majd az ezekhez tartozó saját típusokat is. Ezekből 6db lett, van köztük regex segítségével megadott típus leszűkítés, illetve enumerationnal is. Mindezek után létrehoztam magát a felépítést, amiben az egyszerű típusoknál referáltam a már korábban létrehozottakra, illetve adtam meg minimum és maximum előfordulást is, hiszen az XML-ben egy többszöri előfordulású elementhez, legalább 3 példány tartozik. Minden egyes complexType után megadtam az attribútumokat is, amikhez később létrehoztam a kulcs, idegenkulcs, illetve 1-1 különleges kapcsolat referenciákat is.

Az XMLSchema kódja

```

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

    <!--Egyszerű elemek-->

    <xs:element name="folyometer" type="xs:int" />
    <xs:element name="tipus" type="nyomtatotipTipus" />

```

```

<xs:element name="festekfelhasznalas" type="felhasznTipusok" />
<xs:element name="raszterhenger" type="raszterhengerTipus" />

<xs:element name="lejarat" type="idoTipus" />
<xs:element name="raktaron" type="felhasznTipusok" />
<xs:element name="gyarto" type="xs:string" />

<xs:element name="termeknev" type="xs:string" />
<xs:element name="mennyiseg" type="felhasznTipusok" />
<xs:element name="papirmin" type="felhasznTipusok" />

<xs:element name="nev" type="xs:string" />
<xs:element name="szarmazas" type="xs:string" />
<xs:element name="isz" type="xs:int" />
<xs:element name="utca" type="xs:string" />

<xs:element name="telephely" type="xs:string" />
<xs:element name="alapitas" type="xs:int" />

<xs:element name="elkeszites" type="idoTipus" />
<xs:element name="szinek" type="xs:string" />
<xs:element name="tervezo" type="xs:string" />

<xs:element name="szuletes" type="xs:int" />
<xs:element name="szerzodes" type="xs:int" />

<xs:element name="rendeles" type="idoTipus" />

<!--Saját típusok-->

<xs:simpleType name="nyomtatotipTipus">
  <xs:restriction base="xs:string">
    <xs:enumeration value="VF001" />
    <xs:enumeration value="VF002" />
    <xs:enumeration value="VF003" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="raszterhengerTipus">
  <xs:restriction base="xs:string">
    <xs:pattern value="[1][0-9][0-9]/[0-9]+" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="idoTipus">
  <xs:restriction base="xs:string">
    <xs:pattern value="([12]\d{3}).(0[1-9]|1[0-2]).(0[1-9]|12)\d{3}[01])" />
  </xs:restriction>
</xs:simpleType>

```

```

<xs:simpleType name="felhasznTipusok">
  <xs:restriction base="xs:string">
    <xs:pattern value="[A-Za-z0-9]+" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="adoszamTipus">
  <xs:restriction base="xs:string">
    <xs:pattern value="[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9][0-9][0-9][0-9][0-9]" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="tajszamTipus">
  <xs:restriction base="xs:string">
    <xs:pattern value="[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]" />
  </xs:restriction>
</xs:simpleType>

<!--Felépítés-->

<xs:element name="tasakgyartas">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="nyomtatogep" minOccurs="1" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="folyometer" />
            <xs:element ref="tipus" />
            <xs:element ref="festekfelhasznalas" />
            <xs:element ref="raszterhenger" minOccurs="1" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="termeknev" type="xs:string" />
        </xs:complexType>
      </xs:element>
      <xs:element name="festek" minOccurs="1" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="lejarat" />
            <xs:element ref="raktaron" />
            <xs:element ref="gyarto" minOccurs="1" maxOccurs="unbounded" />
          </xs:sequence>
          <xs:attribute name="pantonszam" type="felhasznTipusok" />
          <xs:attribute name="felhasznalas" type="xs:string" />
        </xs:complexType>
      </xs:element>
      <xs:element name="tasakrendeles" minOccurs="1" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>

```

```

        <xs:element ref="termeknev" />
        <xs:element ref="mennyiseg" />
        <xs:element ref="papirmin" />
    </xs:sequence>
    <xs:attribute name="tasakID" type="xs:int" />
</xs:complexType>
</xs:element>
<xs:element name="vevo" minOccurs="1" maxOccurs="unbounded">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="nev" />
            <xs:element ref="szarmazas" />
            <xs:element name="telephely" minOccurs="1" maxOccurs="2">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element ref="isz" />
                        <xs:element ref="utca" />
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
        <xs:attribute name="adoszam" type="adoszamTipus" />
    </xs:complexType>
</xs:element>
<xs:element name="gyar" minOccurs="1" maxOccurs="unbounded">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="nev" />
            <xs:element ref="telephely" />
            <xs:element ref="alapitas" />
        </xs:sequence>
        <xs:attribute name="gyAdoszam" type="adoszamTipus" />
        <xs:attribute name="vGy" type="adoszamTipus" />
    </xs:complexType>
</xs:element>
<xs:element name="logo" minOccurs="1" maxOccurs="unbounded">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="elkeszites" />
            <xs:element ref="szinek" minOccurs="1" maxOccurs="unbounded" />
            <xs:element ref="tervezo" />
        </xs:sequence>
        <xs:attribute name="logoId" type="xs:int" />
        <xs:attribute name="gyL" type="adoszamTipus" />
    </xs:complexType>
</xs:element>
<xs:element name="tulajdonos" minOccurs="1" maxOccurs="unbounded">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="nev" />

```

```

        <xs:element ref="szuletes" />
        <xs:element ref="szerzodese" />
    </xs:sequence>
    <xs:attribute name="tajszam" type="tajszamTipus" />
    <xs:attribute name="tGy" type="adoszamTipus" />
</xs:complexType>
</xs:element>
<xs:element name="szerzodes" minOccurs="1" maxOccurs="unbounded">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="rendeles" />
        </xs:sequence>
        <xs:attribute name="tNev" type="xs:string" />
        <xs:attribute name="tasakID" type="xs:int" />
        <xs:attribute name="adoszam" type="adoszamTipus" />
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>

<!--Kulcsok-->

<xs:key name="nyomtatogep_kulcs">
    <xs:selector xpath="nyomtatogep" />
    <xs:field xpath="@termeknev" />
</xs:key>

<xs:key name="festek_kulcs">
    <xs:selector xpath="festek" />
    <xs:field xpath="@pantonszam" />
</xs:key>

<xs:key name="tasakrendeles_kulcs">
    <xs:selector xpath="tasakrendeles" />
    <xs:field xpath="@tasakID" />
</xs:key>

<xs:key name="vevo_kulcs">
    <xs:selector xpath="vevo" />
    <xs:field xpath="@adoszam" />
</xs:key>

<xs:key name="gyar_kulcs">
    <xs:selector xpath="gyar" />
    <xs:field xpath="@gyAdoszam" />
</xs:key>

<xs:key name="logo_kulcs">
    <xs:selector xpath="logo" />
    <xs:field xpath="@logoId" />

```

```

</xs:key>

<xs:key name="tulajdonos_kulcs">
  <xs:selector xpath="tulajdonos" />
  <xs:field xpath="@tajszam" />
</xs:key>

<!--Idegen kulcsok-->

<xs:keyref refer="nyomtatogep_kulcs" name="nyomtatogep_idegen_kulcs">
  <xs:selector xpath="festek" />
  <xs:field xpath="@felhasznalas" />
</xs:keyref>

<xs:keyref refer="vevo_kulcs" name="vevo_idegen_kulcs">
  <xs:selector xpath="gyar" />
  <xs:field xpath="@vGy" />
</xs:keyref>

<xs:keyref refer="gyar_kulcs" name="gyar_tulajdonos_idegen_kulcs">
  <xs:selector xpath="tulajdonos" />
  <xs:field xpath="@tGy" />
</xs:keyref>

<xs:keyref refer="nyomtatogep_kulcs" name="szerzodes_nyomtatogep_idegen_kulcs">
  <xs:selector xpath="szerzodes" />
  <xs:field xpath="@tNev" />
</xs:keyref>

<xs:keyref refer="vevo_kulcs" name="szerzodes_vevo_idegen_kulcs">
  <xs:selector xpath="szerzodes" />
  <xs:field xpath="@adoszam" />
</xs:keyref>

<xs:keyref refer="tasakrendeles_kulcs" name="szerzodes_tasakrendeles_idegen_kulcs">
  <xs:selector xpath="szerzodes" />
  <xs:field xpath="@tasakID" />
</xs:keyref>

<!--1:1-->

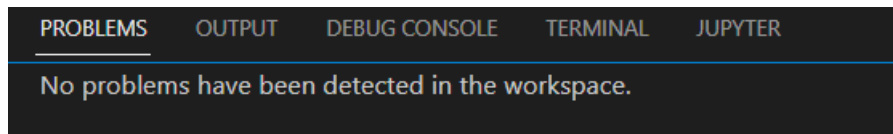
<xs:unique name="unique_logo">
  <xs:selector xpath="logo" />
  <xs:field xpath="@gyL" />
</xs:unique>

</xs:element>

</xs:schema>

```

Validálás sikeressége:



DOM adatolvasás

Először is a java kódban beállítom, hogy melyik az a file, amiből kellene olvasni, ezt a file-t, pedig a projekt mappájában helyeztem el. Létrehozok egy document elemet, amely segítségével le fogom tudni kérdezni a root elementet, illetve a több parent, illetve gyerekelemet is. NodeList-et kell létrehozni, hogy a többszöri előfordulású elementeket el lehessen tárolni, ezeket a document.getElementsByTagName(String) segítségével lehet lekérdezni. Miután ez megtörtént for ciklussal kell végigiterálni az adott parenteleme-n, hogy az attribútumokat és a gyerekelemeket el tudjuk menteni egy-egy stringbe, majd ki tudjuk írni őket. Ahol olyan elem van, aminek több értéke lehet, ott megvizsgálom, hogy az elemek száma több-e, mint akkor, hogy ha nem lenne az elemnek több értéke. Ha igen, akkor ciklussal iterálok, és kiíratom a több értékeket.

Kód:

```
package hu.domparse.DJ7PNE;

import java.io.File;
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.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;

public class DOMReadDJ7PNE {

    public static void main(String argv[]) throws SAXException, IOException,
ParserConfigurationException {

        File xmlFile = new File("XMLDJ7PNE.xml");

        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = factory.newDocumentBuilder();

        Document doc = dBuilder.parse(xmlFile);

        doc.getDocumentElement().normalize();
```

```

        System.out.println("Root element: " +
doc.getDocumentElement().getNodeName());

        NodeList nList = doc.getElementsByTagName("nyomtatogep");

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

            Node nNode = nList.item(i);
            System.out.println("\nCurrent element: " + nNode.getNodeName());

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

                Element elem = (Element) nNode;
                String termeknev = elem.getAttribute("termeknev");

                Node node1 =
elem.getElementsByTagName("folyometer").item(0);
                String fm = node1.getTextContent();

                Node node2 = elem.getElementsByTagName("tipus").item(0);
                String tip = node2.getTextContent();

                Node node3 =
elem.getElementsByTagName("festekfelhasznalas").item(0);
                String ff = node3.getTextContent();

                System.out.println("Termeknev: " + termeknev);
                System.out.println("Felhasznált folyóméter: " + fm);
                System.out.println("Gép típusa: " + tip);
                System.out.println("Nyomtatás során felhasznált
festékmennyiség: " + ff);

                if (nList.item(i).getChildNodes().getLength() > 5) {
                    int db = 0;
                    Node node4 =
elem.getElementsByTagName("raszterhenger").item(0);
                    while (node4 != null) {
                        node4 =
elem.getElementsByTagName("raszterhenger").item(db);
                        if (node4 != null) {
                            String rh = node4.getTextContent();
                            System.out.println("Nyomtatáshoz
felhasznált raszterhenger: " + rh);
                        }
                        db++;
                    }
                }
            }
        }
    }
}

```



```

    }
}

NodeList fList = doc.getElementsByTagName("festek");

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

    Node fNode = fList.item(i);
    System.out.println("\nCurrent element: " + fNode.getNodeName());

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

        Element elem = (Element) fNode;
        String psz = elem.getAttribute("pantonszam");
        String felh = elem.getAttribute("felhasznalas");

        Node node1 = elem.getElementsByTagName("lejarat").item(0);
        String lej = node1.getTextContent();

        Node node2 = elem.getElementsByTagName("raktaron").item(0);
        String rak = node2.getTextContent();

        System.out.println("Pantonszam: " + psz);
        System.out.println("Melyik termékhez van felhasználva: " + felh);

        System.out.println("Lejárat dátum: " + lej);
        System.out.println("Mennyiség ami a raktáron van: " + rak);

        if (fList.item(i).getChildNodes().getLength() > 3) {
            int db = 0;
            Node node3 = elem.getElementsByTagName("gyarto").item(0);
            while (node3 != null) {
                node3 = elem.getElementsByTagName("gyarto").item(db);
                if (node3 != null) {
                    String gy = node3.getTextContent();
                    System.out.println("A festéket gyártja: " + gy);
                }
                db++;
            }
        }

    }

}

NodeList tList = doc.getElementsByTagName("tasakrendeles");

```

```

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

    Node tNode = tList.item(i);
    System.out.println("\nCurrent element: " + tNode.getNodeName());

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

        Element elem = (Element) tNode;
        String tid = elem.getAttribute("tasakID");

        Node node1 = elem.getElementsByTagName("termeknev").item(0);
        String tnev = node1.getTextContent();

        Node node2 = elem.getElementsByTagName("mennyiseg").item(0);
        String mennyi = node2.getTextContent();

        Node node3 = elem.getElementsByTagName("papirmin").item(0);
        String pm = node3.getTextContent();

        System.out.println("Tasak egyedi azonosítója: " + tid);
        System.out.println("Rendelt tasak neve: " + tnev);
        System.out.println("Rendelt mennyiség: " + mennyi);
        System.out.println("Gyártáshoz használt papír minősége: " + pm);

    }

}

NodeList vList = doc.getElementsByTagName("vevo");

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

    Node vNode = vList.item(i);
    System.out.println("\nCurrent element: " + vNode.getNodeName());

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

        Element elem = (Element) vNode;
        String asz = elem.getAttribute("adoszam");

        Node node1 = elem.getElementsByTagName("nev").item(0);
        String nev = node1.getTextContent();

        Node node2 = elem.getElementsByTagName("szarmazas").item(0);
        String szarm = node2.getTextContent();
    }

}

```

```

        System.out.println("Vevő adószáma: " + asz);
        System.out.println("Rendelő neve: " + nev);
        System.out.println("Rendelt mennyiség: " + szarm);

        if (fList.item(i).getChildNodes().getLength() > 3) {
            int db = 0;
            Node node3 =
elem.getElementsByTagName("telephely").item(0);
            while (node3 != null) {
                node3 =
elem.getElementsByTagName("telephely").item(db);
                if (node3 != null) {
                    Node n =
elem.getElementsByTagName("isz").item(db);
                    String isz = n.getTextContent();
                    System.out.println("Telephely
irányítószáma: " + isz);
                    Node n2 =
elem.getElementsByTagName("utca").item(db);
                    String u = n2.getTextContent();
                    System.out.println("Telephely utcája: " +
u);
                }
                db++;
            }
        }
    }
}

NodeList gyList = doc.getElementsByTagName("gyar");

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

    Node gyNode = gyList.item(i);
    System.out.println("\nCurrent element: " + gyNode.getNodeName());

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

        Element elem = (Element) gyNode;
        String gyasz = elem.getAttribute("gyAdoszam");
        String vgy = elem.getAttribute("vGy");

        Node node1 = elem.getElementsByTagName("nev").item(0);
        String nev = node1.getTextContent();

        Node node2 =
elem.getElementsByTagName("telephely").item(0);

```

```

        String telep = node2.getTextContent();

        Node node3 = elem.getElementsByTagName("alapitas").item(0);
        String ap = node3.getTextContent();

        System.out.println("Gyár adószáma: " + gyasz);
        System.out.println("Vevő adószáma, akinek a rendelése itt
készül: " + vgy);

        System.out.println("A gyár neve: " + nev);
        System.out.println("A telephely városa: " + telep);
        System.out.println("A gyár alapításának az éve: " + ap);

    }
}

NodeList lList = doc.getElementsByTagName("logo");

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

    Node lNode = lList.item(i);
    System.out.println("\nCurrent element: " + lNode.getNodeName());

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

        Element elem = (Element) lNode;
        String lid = elem.getAttribute("logoId");
        String gyl = elem.getAttribute("gyL");

        Node node1 = elem.getElementsByTagName("elkeszites").item(0);
        String elk = node1.getTextContent();

        Node node3 = elem.getElementsByTagName("tervezo").item(0);
        String terv = node3.getTextContent();

        System.out.println("A logo egyedi azonosítója: " + lid);
        System.out.println("A gyár adószáma, aminek a logója ez: " +
gyl);

        System.out.println("A logo elkészülésének időpontja: " + elk);

        if (lList.item(i).getChildNodes().getLength() > 2) {
            int db = 0;
            Node node2 = elem.getElementsByTagName("szinek").item(0);
            while (node2 != null) {
                node2 = elem.getElementsByTagName("szinek").item(db);
                if (node2 != null) {
                    String szin = node2.getTextContent();

```

```

        szín: " + szín);
        }
        db++;
    }

    System.out.println("A logo tervezője: " + terv);
}

NodeList tnList = doc.getElementsByTagName("tulajdonos");

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

    Node tnNode = tnList.item(i);
    System.out.println("\nCurrent element: " + tnNode.getNodeName());

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

        Element elem = (Element) tnNode;
        String tsz = elem.getAttribute("tajszam");
        String tgy = elem.getAttribute("tGy");

        Node node1 = elem.getElementsByTagName("nev").item(0);
        String nev = node1.getTextContent();

        Node node2 = elem.getElementsByTagName("szuletes").item(0);
        String szul = node2.getTextContent();

        Node node3 = elem.getElementsByTagName("szerzodes").item(0);
        String szer = node3.getTextContent();

        System.out.println("Tulajdonos tajszáma: " + tsz);
        System.out.println("A gyár adószáma, aminek a tulajja az ember: " + tgy);

        System.out.println("A tulajdonos neve: " + nev);
        System.out.println("A tulajdonos születése: " + szul);
        System.out.println("Mióta szerződött ide: " + szer);

    }

}

NodeList szerList = doc.getElementsByTagName("szerzodes");

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

```

```

Node szerNode = szerList.item(i);
System.out.println("\nCurrent element: " + szerNode.getNodeName());

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

    Element elem = (Element) szerNode;

    String tnev = elem.getAttribute("tNev");
    String tid = elem.getAttribute("tasakID");
    String asz = elem.getAttribute("adoszam");

    Node node1 = elem.getElementsByTagName("rendeles").item(0);
    String rend = node1.getTextContent();

    System.out.println("Megrendelt termék neve: " + tnev);
    System.out.println("A tasak ID-ja, ami meg lett rendelve: " + tid);
    System.out.println("A vevő adószáma, aki rendelt: " + asz);
    System.out.println("Rendelés időpontja: " + rend);

}

}

}

```

DOM adatmódosítás

Kiválasztottam, hogy az első nyomtatógép attribútumát szeretném módosítani, illetve egy gyerekelemét a folyómétert. Mivel ez az attribútum több helyen is megjelent, pl a festék, tasakrendelés, illetve szerződés elemeknél is, ezért ezeket ott is meg kellett változtatnom, illetve, mivel a folyóméter nagyságát is változtattam, ezért a rendelt mennyiséget is meg kellett változtatni a tasakrendelés elemnél. A módosított fájlt mentettem, illetve kiíratam a konzolra.

Kód:

```

package hu.domparse.DJ7PNE;

import java.io.File;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;

import org.w3c.dom.Document;

```

```

import org.w3c.dom.Element;
import org.w3c.dom.NamedNodeMap;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;

public class DOMModifyDJ7PNE {

    public static void main(String argv[]) {

        try {

            File inputFile = new File("XMLDJ7PNE2.xml");

            DocumentBuilderFactory documentBuilderFactory =
DocumentBuilderFactory.newInstance();
            DocumentBuilder documentBuilder =
documentBuilderFactory.newDocumentBuilder();

            Document doc = documentBuilder.parse(inputFile);

            //nyomtatógép attribútumának módosítása
            Node nyomtatog =
doc.getElementsByTagName("nyomtatogep").item(0);

            NamedNodeMap attr = nyomtatog.getAttributes();
            Node nodeAttr = attr.getNamedItem("termeknev");
            nodeAttr.setTextContent("1kg Nagyí Titka");

            //nyomtatógép folyóméter attribútiumjának módosítása
            NodeList list = nyomtatog.getChildNodes();

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

                if (node.getNodeType() == Node.ELEMENT_NODE) {
                    Element eElement = (Element) node;

                    if ("folyometer".equals(eElement.getNodeName())) {
                        if ("2500".equals(eElement.getTextContent())) {
                            eElement.setTextContent("3200");
                        }
                    }
                }
            }

            //festék attribútumának módosítása
            NodeList festek = doc.getElementsByTagName("festek");

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

```

```

        if (node.getNodeType() == Node.ELEMENT_NODE) {
            Element eElement = (Element) node;

            if ("1kg Gyermelyi
finomliszt".equals(eElement.getAttribute("felhasznalas"))) {
                NamedNodeMap attr2 = eElement.getAttributes();
                Node nodeAttr2 =
attr2.getNamedItem("felhasznalas");
                nodeAttr2.setTextContent("1kg Nagyí Titka");
            }
        }
    }

    //Tasakrendelésben való módosítás
    Node trr = doc.getElementsByTagName("tasakrendeles").item(0);
    NodeList tr = trr.getChildNodes();

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

        if (node.getNodeType() == Node.ELEMENT_NODE) {
            Element eElement = (Element) node;

            if ("termeknev".equals(eElement.getNodeName())) {
                if ("1kg Gyermelyi
finomliszt".equals(eElement.getTextContent())) {
                    eElement.setTextContent("1kg Nagyí
Titka");
                }
            }
            if ("mennyiseg".equals(eElement.getNodeName())) {
                if ("150000db".equals(eElement.getTextContent())) {
                    eElement.setTextContent("180000db");
                }
            }
        }
    }

    //Szerződés kapcsolatban való módosítás

    NodeList sz = doc.getElementsByTagName("szerzodes");

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

        if (node.getNodeType() == Node.ELEMENT_NODE) {
            Element eElement = (Element) node;

```



```

        if ("1kg Gyermelyi
finomliszt".equals(eElement.getAttribute("tNev"))) {
            NamedNodeMap attr2 = eElement.getAttributes();
            Node nodeAttr2 = attr2.getNamedItem("tNev");
            nodeAttr2.setTextContent("1kg Nagy Titka");
        }
    }

    //Tartalom konzolra, illetve fájlba való írása

    TransformerFactory transformerFactory =
TransformerFactory.newInstance();
    Transformer transformer = transformerFactory.newTransformer();

    DOMSource source = new DOMSource(doc);

    System.out.println("-----Módosított fájl-----");

    StreamResult consoleResult = new StreamResult(System.out);
    StreamResult file = new StreamResult(inputFile);

    transformer.transform(source, consoleResult);
    transformer.transform(source, file);

    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

DOM adatlekérdezés

A lekérdezéseket az XPath segítségével hajtottam végre, ebből 9-et csináltam, majd ezeket teszteltem, a kritériumoknak megfelelőket kiírtattam a konzolra.

Kód:

```

package hu.domparse.DJ7PNE;

import java.io.IOException;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.xpath.XPath;
import javax.xml.xpath.XPathConstants;
import javax.xml.xpath.XPathExpressionException;

```

```

import javax.xml.xpath.XPathFactory;

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

public class DOMQueryDJ7PNE {

    public static void main(String[] args) {

        try {

            // DocumentBuilder létrehozása
            DocumentBuilderFactory documentBuilderFactory =
DocumentBuilderFactory.newInstance();
            DocumentBuilder documentBuilder =
documentBuilderFactory.newDocumentBuilder();

            Document document = documentBuilder.parse("XMLDJ7PNE.xml");

            document.getDocumentElement().normalize();

            // az XPath készítése
            XPath xPath = XPathFactory.newInstance().newXPath();

            // meg kell adni az elérési út kifejezést és a csomópont listát

            // a tasakgyartas root elem nyomtatogep gyerekelemeinek lekérdezése
            String expression = "tasakgyartas / nyomtatogep";

            // 2-es ID-jű tasakrendelés lekérdezése
            // String expression = "//tasakrendeles[@tasakID='2']";

            // a második vevő kiválasztása
            // String expression = "tasakgyartas/vevo[2]";

            // a logók, amikben van zöld szín felhasználva
            // String expression = "//logo[szinek='Zöld']";

            // a logóknak, amelyekben van zöld szín, azoknak az elkészítési ideje
            // String expression = "//logo[szinek='Zöld']/elkeszites";

            // Logo elkészítési időpontja, és tervező kiírása
            // String expression = "//elkeszites | //tervezo";

```

```
// Szerződés kapcsolat első két eleme
// String expression = "//szerzodes[position()<3]";

// Tasakrendelés papírminőségei, ahol a mennyiség több, mint 80000db
//String expression = "//tasakrendeles[mennyiseg>80000]/papirmin";

//Vevők akik rendelkeznek bármilyen attribútummal
//String expression = "//vevo[@*]";
```

értékelni // Készítünk egy listát, majd az XPath kifejezést le kell fordítani és ki kell

```
NodeList nodeList = (NodeList)
XPath.compile(expression).evaluate(document, XPathConstants.NODESET);
```

```
// A for ciklus segítségével a NodeList csomópontjain végig kell iteralni
for (int i = 0; i < nodeList.getLength(); i++) {
    Node node = nodeList.item(i);
```

```
System.out.println("\nAktuális elem: " + node.getNodeName());
```

// Meg kell vizsgálni a csomópontot, tesztelni kell a subelementet, jelen esetben a nyomtatógép elemet

```
if (node.getNodeType() == Node.ELEMENT_NODE &&
node.getNodeName().equals("nyomtatogep")) {
```

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

```
System.out.println("Terméknév: " +
element.getAttribute("termeknev"));
```

```
System.out.println(
"Folyóméter: " +
element.getElementsByTagName("folyometer").item(0).getTextContent());
```

```
System.out.println("Típus: " +
element.getElementsByTagName("tipus").item(0).getTextContent());
```

```
System.out.println("Festékfelhasználás: "
+
element.getElementsByTagName("festekfelhasznalas").item(0).getTextContent());
```

```
if (nodeList.item(i).getChildNodes().getLength() > 5) {
    int db = 0;
    Node node4 =
element.getElementsByTagName("raszterhenger").item(0);
    while (node4 != null) {
```

```

                                node4
                                =
element.getElementsByTagName("raszterhenger").item(db);
                                if (node4 != null) {
                                    String rh =
node4.getTextContent();
                                System.out.println("Nyomtatáshoz
felhasznált raszterhenger: " + rh);
                                }
                                db++;
                            }
                        }

// Logo elkészítés kiíratása
if (node.getNodeType() == Node.ELEMENT_NODE &&
node.getNodeName().equals("elkeszites")) {

    Element element = (Element) node;

    System.out.println("Idopont: " +
element.getTextContent());

}

// Logo tervezőjének kiíratása
if (node.getNodeType() == Node.ELEMENT_NODE &&
node.getNodeName().equals("tervezo")) {

    Element element = (Element) node;

    System.out.println("Tervezo: " +
element.getTextContent());

}

// Tasakrendelés papírminőségének kiíratása
if (node.getNodeType() == Node.ELEMENT_NODE &&
node.getNodeName().equals("papirmin")) {

    Element element = (Element) node;

    System.out.println("Papírminőség: " +
element.getTextContent());

}

//Tasakrendelés kiíratása

```



```

        System.out.println("Telephely
        irányítószáma: " + isz);
        element.getElementsByTagName("utca").item(db);
        Node n2 =
        String u = n2.getTextContent();
        System.out.println("Telephely
        utcája: " + u);
    }
    db++;
}

}

}

//Logo kiíratása
if (node.getNodeType() == Node.ELEMENT_NODE &&
node.getNodeName().equals("logo")) {
    Element element = (Element) node;
    System.out.println("logID: " +
element.getAttribute("logoId"));
    System.out.println("Gyár adószáma, akié a logo: " +
element.getAttribute("gyL"));
    System.out.println(
        "Elkészítés: " +
element.getElementsByTagName("elkeszites").item(0).getTextContent());
    if (nodeList.item(i).getChildNodes().getLength() > 2) {
        int db = 0;
        Node node2 =
element.getElementsByTagName("szinek").item(0);
        while (node2 != null) {
            node2 =
element.getElementsByTagName("szinek").item(db);
            if (node2 != null) {
                String szin =
node2.getTextContent();
                System.out.println("A logohoz
                felhasznált szín: " + szin);
            }
            db++;
        }
    }
}

```

```

        if (node.getNodeType() == Node.ELEMENT_NODE
&& node.getNodeName().equals("szinek")) {
            System.out.println("A logóhoz felhasznált szín: "
+
element.getElementsByTagName("szinek").item(0).getTextContent());
        }

    }

    //Szerződés kiírása
    if (node.getNodeType() == Node.ELEMENT_NODE &&
node.getNodeName().equals("szerzodes")) {

        Element element = (Element) node;

        System.out.println("tNev: " +
element.getAttribute("tNev"));

        System.out.println("tasakID: " +
element.getAttribute("tasakID"));

        System.out.println("Adószám: " +
element.getAttribute("adoszam"));

        System.out
            .println("Rendelés: " +
element.getElementsByTagName("rendeles").item(0).getTextContent());

    }

}

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

}

}

```