

**Miskolci Egyetem**

Gépészmérnöki és Informatikai Kar

Jegyzőkönyv  
Adatkezelés XML környezetben  
Féléves feladat  
Videótéka

Gáspárfalvi Péter

WUWIOS

# Tartalomjegyzék

1. Feladat leírása: .....	1
2. Feladat első része .....	1
2.1. ER modell .....	1
2.2. XDM modellre konvertálás.....	3
2.3. XML dokumentum.....	3
2.4. XMLSchema .....	7
3. Feladat második része .....	14
3.1. Adatbázis beolvasása DOM segítségével .....	14
3.2. Adatbázis módosítása DOM segítségével.....	19
3.3. Adatbázis lekérdezése DOM segítségével .....	24

## 1. Feladat leírása

A féléves feladat két részből áll. Az első feladatrészben a kiválasztott és oktatóval egyeztetett témaválasztásnak megfelelően kell létrehozni egy XML adatkezelő rendszert. Az adatbázis ER modellje tartalmazzon minimum 5 egyedet, többféle kapcsolatot és minden fajtájú tulajdonságot. Az ER modell egyes egyedeinek minimum 4 tulajdonsága legyen és egy N:M kapcsolatnak is legyenek tulajdonságai. Az ER modellnek megfelelően létre kell hozni egy XDM modellt, melynek alapján egy XML dokumentumot és hozzá tartozó XMLSchema-t kell elkészíteni. Az XMLSchema-ban legyen saját típus. A feladat második részében egy DOM programot kell írni Java környezetben. Három Java class-t kell létrehozni: adatolvasásra, adatlekérdezésre és adatmódosítást elvégzőt. A classokat elvárt módon packagelni kell és a kódot kommentekkel kell ellátni.

## 2. Feladat első része

### 2.1. ER modell

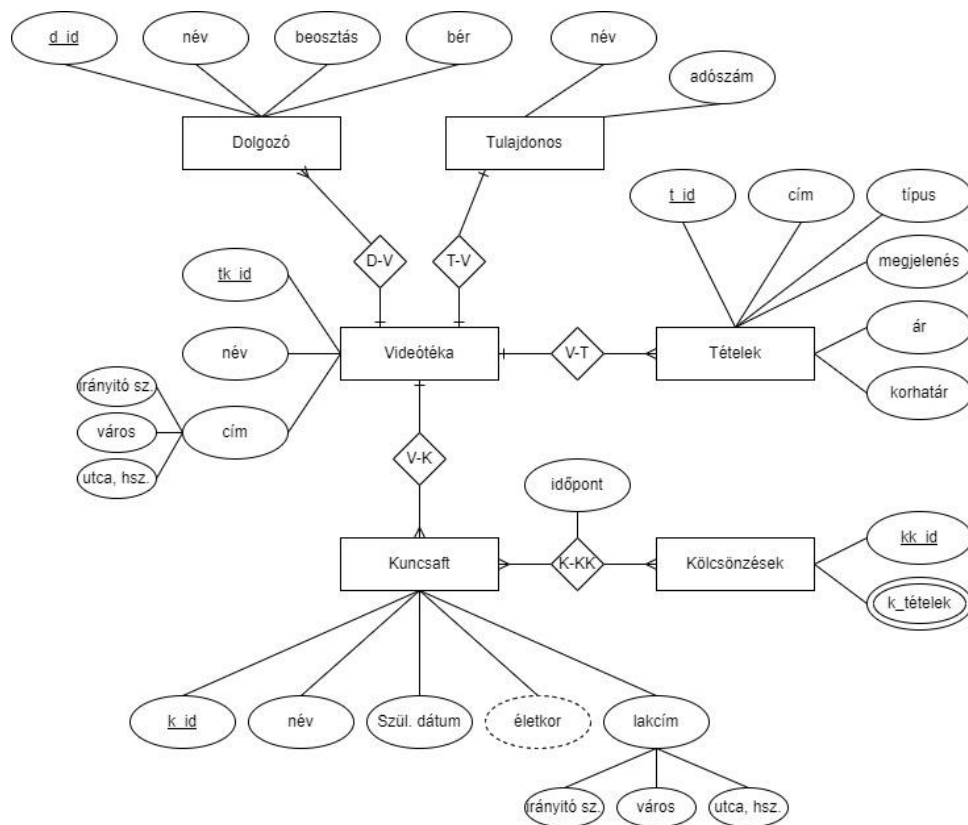
ER modellt készítek, ami így írható le egyedei és kapcsolatai alapján:

Egyedek:

- Videótéka: Videótéka egyedi azonosítóját 'tk\_id', a videótéka nevét és címét tárolja el. A cím összetett tulajdonság: irányítószám, város, és utca-házzám tulajdonságokból áll.
- Tulajdonos: A videótéka tulajdonosának neve és adószáma.
- Dolgozó: A videótékában dolgozók azonosítóját 'd\_id', nevét, beosztását és bérét tartalmazó egyed.
- Tételek: A videótékában készleten tartott DVD-k és Kazettákat tartalmazó egyed. Egyedi azonosító tulajdonsága a 't\_id', ezenkívül a filmek címeit és azt, hogy DVD vagy Kazetta a 'típus' tulajdonság írja le. A megjelenés dátumát az árat és a korhatárt tárolja.
- Kuncsaft: A videótékából kölcsönző személyek adatait leíró egyed. Tartalmaz származtatott tulajdonságot: 'életkor', amit a születési dátum tulajdonság alapján lehet megadni.
- Kölcsönzések: A kuncsaftok által kikölcsönzött tételeket többértékű tulajdonságban tartalmazza.

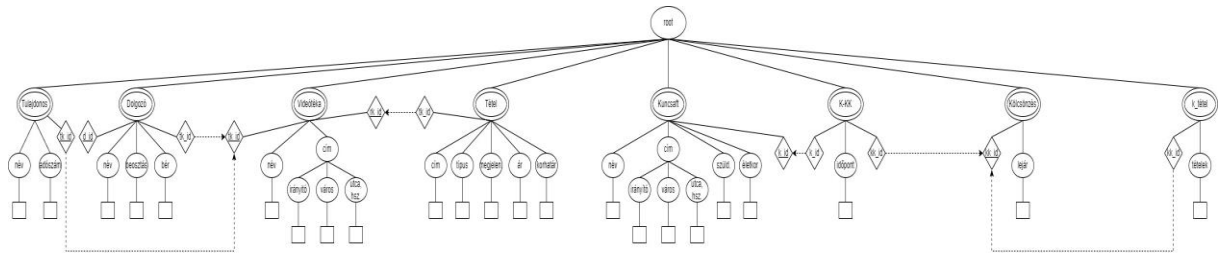
Kapcsolatok:

- D-V: 1:N kapcsolat a dolgozók és videótéka között.
- T-V: 1:1 kapcsolat tulajdonos és a videótéka között.
- V-K: 1:N kapcsolat a videótéka és a kuncsaftjai között.
- V-T: 1:N kapcsolat a videótéka és a tételei között.
- K-KK: N:M kapcsolat a kuncsaftok és a kölcsönzések között. Tulajdonsága is van a kapcsolatnak: 'időpont'.



## 2.2. XDM modellre konvertálás

ER modellt XDM modellre úgy konvertálhatunk, hogy háromféle jelölést használunk. Van az elem, amelyet az XDM modellben ellipszissel lehet ábrázolni. Az attribútumot rombuszsal lehet jelölni, illetve a szöveget a téglalapban. Ezeken felül van még a dupla ellipszis, amely a többszörös előfordulási lehetőséget határozza meg. A származtatott tulajdonságok elhagyhatóak.



## 2.3. XML dokumentum

```
<?xml version="1.0" encoding="UTF-8"?>
<root xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="XMLSchemaWUWIOS.xsd">
  <!-- videotéka létrehozása -->
    <videoteka tk_id = "1">
      <nev>Menő TK</nev>
      <cim>
        <iranyitosz>3535</iranyitosz>
        <varos>Miskolc</varos>
        <utcahsz>Eper 2.</utcahsz>
      </cim>
    </videoteka>
  <!-- a dolgozók létrehozása -->
    <dolgozo d_id = "1" tk_id="1">
      <nev>Cserepes Virág</nev>
      <beosztas>vezető</beosztas>
      <ber>240000</ber>
    </dolgozo>
    <dolgozo d_id = "2" tk_id="1">
      <nev>Ebéd Elek</nev>
      <beosztas>eladó</beosztas>
      <ber>180000</ber>
    </dolgozo>
```

```
<dolgozo d_id = "3" tk_id="1">
  <nev>Rönk Ödön</nev>
  <beosztas>eladó</beosztas>
  <ber>190000</ber>
</dolgozo>
<!-- a tételek létrehozása -->
<tétel t_id = "1" tk_id = "1">
  <cím>Halálos iramban</cím>
  <tipus>Kazetta</tipus>
  <megjelenes>2001.06.22.</megjelenes>
  <ar>1000</ar>
  <korhatar>16</korhatar>
</tétel>
<tétel t_id = "2" tk_id = "1">
  <cím>Halálosabb iramban</cím>
  <tipus>Kazetta</tipus>
  <megjelenes>2003.06.06.</megjelenes>
  <ar>1200</ar>
  <korhatar>12</korhatar>
</tétel>
<tétel t_id = "3" tk_id = "1">
  <cím>Halálos iramban: Tokiói hajsza</cím>
  <tipus>DVD</tipus>
  <megjelenes>2006.06.04.</megjelenes>
  <ar>1400</ar>
  <korhatar>12</korhatar>
</tétel>
<tétel t_id = "4" tk_id = "1">
  <cím>Halálos iram</cím>
  <tipus>DVD</tipus>
  <megjelenes>2009.04.03.</megjelenes>
  <ar>1400</ar>
  <korhatar>16</korhatar>
</tétel>
<tétel t_id = "5" tk_id = "1">
  <cím>Halálos iram: Ötödik sebesség</cím>
  <tipus>DVD</tipus>
  <megjelenes>2011.04.29.</megjelenes>
```

```
<ar>1600</ar>
<korhatar>16</korhatar>
</tetel>
<tetel t_id = "6" tk_id = "1">
  <cim>Halálos iramban 6</cim>
  <tipus>DVD</tipus>
  <megjelenes>2013.05.24.</megjelenes>
  <ar>1600</ar>
  <korhatar>16</korhatar>
</tetel>
<tetel t_id = "7" tk_id = "1">
  <cim>Halálos iramban 7</cim>
  <tipus>DVD</tipus>
  <megjelenes>2015.04.03.</megjelenes>
  <ar>1600</ar>
  <korhatar>16</korhatar>
</tetel>
<!-- a kuncsaftok létrehozása -->
<kuncsaft k_id = "1" tk_id = "1">
  <nev>Nagy Endre</nev>
  <szuld>1987.01.07.</szuld>
  <lakcim>
    <iranyitosz>3535</iranyitosz>
    <varos>Miskolc</varos>
    <utcahsz>Béke utca 2.</utcahsz>
  </lakcim>
</kuncsaft>
<kuncsaft k_id = "2" tk_id = "1">
  <nev>Kiss Virág</nev>
  <szuld>2004.05.07.</szuld>
  <lakcim>
    <iranyitosz>3535</iranyitosz>
    <varos>Miskolc</varos>
    <utcahsz>Békés utca 8.</utcahsz>
  </lakcim>
</kuncsaft>
<kuncsaft k_id = "3" tk_id = "1">
  <nev>Közepes Kálmán</nev>
```

```

        <szuld>2000.05.09.</szuld>
        <lakcim>
            <iranyitosz>3535</iranyitosz>
            <varos>Miskolc</varos>
            <utcahsz>Békétlen utca 10.</utcahsz>
        </lakcim>
    </kuncsaft>
<!-- a kölcsönzések létrehozása -->
    <kolcsonzes kk_id = "1">
        <tétel>
            <t>1</t>
        </tétel>
    </kolcsonzes>
    <kolcsonzes kk_id = "2">
        <tétel>
            <t>2</t>
        </tétel>
    </kolcsonzes>
    <kolcsonzes kk_id = "3">
        <tétel>
            <t>5</t>
            <t>6</t>
        </tétel>
    </kolcsonzes>
<!-- a kuncsaft kölcsönzések kapcsoló tábla létrehozása -->
    <k_kk k_id= "1" kk_id = "1">
        <idopont>2022.05.06.</idopont>
    </k_kk>
    <k_kk k_id= "2" kk_id = "1">
        <idopont>2022.05.06.</idopont>
    </k_kk>
    <k_kk k_id= "3" kk_id = "2">
        <idopont>2022.07.20.</idopont>
    </k_kk>
    <k_kk k_id= "2" kk_id = "3">
        <idopont>2022.03.10.</idopont>
    </k_kk>
</root>

```



## 2.4. XMLSchema

Az XML dokumentumhoz létre kell hoznom egy, a validációt lehetővé tevő sémát. Egyszerű elemeket veszem fel először melyeket majd később referálok. Létrehozom a megfelelő saját típusokat, kulcsokat és kulcs refet. Az 1:1 kapcsolat biztosítására unique feltételt alkalmazok.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="nev" type="xs:string" />
  <xs:element name="iranyitosz" type="iranyitoszType" />
  <xs:element name="varos" type="xs:string" />
  <xs:element name="utcahsz" type="xs:string" />
  <xs:element name="adoszam" type="xs:string" />
  <xs:element name="beosztas" type="beosztasType" />
  <xs:element name="ber" type="xs:positiveInteger" />
  <xs:element name="cim" type="xs:string" />
  <xs:element name="tipus" type="tipusType" />
  <xs:element name="megjelenes" type="datumType" />
  <xs:element name="ar" type="xs:positiveInteger" />
  <xs:element name="korhatar" type="xs:string" />
  <xs:element name="szuld" type="datumType" />
  <xs:element name="idopont" type="datumType" />
  <xs:element name="root">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="videoteka" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element ref="nev" />
              <xs:element name="cim">
                <xs:complexType>
```

```

        <xs:sequence>
            <xs:element ref="iranyitosz" />
            <xs:element ref="varos" />
            <xs:element ref="utcahsz" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="tk_id" type="xs:integer"
use="required" />
</xs:complexType>
</xs:element>
<xs:element name="tulajdonos" maxOccurs="unbounded">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="nev" />
            <xs:element ref="adoszam">
                </xs:element>
            </xs:sequence>
            <xs:attribute name="tk_id" type="xs:integer"
use="required" />
        </xs:complexType>
    </xs:element>
<xs:element name="dolgozo" maxOccurs="unbounded">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="nev"/>
            <xs:element ref="beosztas" />
            <xs:element ref="ber"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

```

        <xs:attribute name="d_id" type="xs:integer"
use="required" />

        <xs:attribute name="tk_id" type="xs:integer"
use="required" />

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

<xs:element name="tetel" maxOccurs="unbounded">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="cim" />
            <xs:element ref="tipus" />
            <xs:element ref="megjelenes" />
            <xs:element ref="ar" />
            <xs:element ref="korhatar" />
        </xs:sequence>
        <xs:attribute name="t_id" type="xs:integer"
use="required" />
        <xs:attribute name="tk_id" type="xs:integer"
use="required" />
    </xs:complexType>
</xs:element>

<xs:element name="kuncsaft" maxOccurs="unbounded">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="nev" />
            <xs:element ref="szuld" />
            <xs:element name="lakcim">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element ref="iranyitosz" />

```

```

        <xs:element ref="varos" />
        <xs:element ref="utcahsz" />
    </xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="k_id" type="xs:integer"
use="required" />
<xs:attribute name="tk_id" type="xs:integer"
use="required" />
</xs:complexType>
</xs:element>
<xs:element name="kolcsonzes" maxOccurs="unbounded">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="tételék">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="t" minOccurs="1"
maxOccurs="unbounded" type="kolcsonzesType" />
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
        <xs:attribute name="kk_id" type="xs:integer"
use="required" />
    </xs:complexType>
</xs:element>
<xs:element name="k_kk" maxOccurs="unbounded">
    <xs:complexType>

```

```

        <xs:sequence>
            <xs:element ref="idopont" />
        </xs:sequence>
        <xs:attribute name="k_id" type="xs:integer"
use="required" />
        <xs:attribute name="kk_id" type="xs:integer"
use="required" />
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
<xs:key name="videoteka_key">
    <xs:selector xpath="videoteka"/>
    <xs:field xpath="@tk_id"/>
</xs:key>
<xs:key name="dolgozo_key">
    <xs:selector xpath="dolgozo"/>
    <xs:field xpath="@d_id"/>
</xs:key>
<xs:key name="tetel_key">
    <xs:selector xpath="tetel"/>
    <xs:field xpath="@t_id"/>
</xs:key>
<xs:key name="kunsaft_key">
    <xs:selector xpath="kunsaft"/>
    <xs:field xpath="@k_id"/>
</xs:key>
<xs:key name="kolcsonzes_key">
    <xs:selector xpath="kolcsonzes"/>
    <xs:field xpath="@kk_id"/>

```

```

</xs:key>

<xs:keyref name="Fkey_dolgozo_tk" refer="videoteka_key">
    <xs:selector xpath="dolgozo"/>
    <xs:field xpath="@tk_id"/>
</xs:keyref>

<xs:keyref name="Fkey_tetel_tk" refer="videoteka_key">
    <xs:selector xpath="tetel"/>
    <xs:field xpath="@tk_id"/>
</xs:keyref>

<xs:keyref name="Fkey_kunsaft_tk" refer="videoteka_key">
    <xs:selector xpath="kunsaft"/>
    <xs:field xpath="@tk_id"/>
</xs:keyref>

<xs:keyref name="Fkey_kapcs_kunsaft" refer="kunsaft_key">
    <xs:selector xpath="k_kk"/>
    <xs:field xpath="@k_id"/>
</xs:keyref>

<xs:keyref name="Fkey_kapcs_kolcsonzes" refer="kolcsonzes_key">
    <xs:selector xpath="k_kk"/>
    <xs:field xpath="@kk_id"/>
</xs:keyref>

<xs:keyref name="Fkey_tk_tulaj" refer="videoteka_key">
    <xs:selector xpath="tulajdonos"/>
    <xs:field xpath="@tk_id"/>
</xs:keyref>

<xs:unique name="unique_fonok">
    <xs:selector xpath="tulajdonos" />
    <xs:field xpath="@tk_id" />
</xs:unique>

```



```
<xs:enumeration value="1"/>
<xs:enumeration value="2"/>
<xs:enumeration value="3"/>
<xs:enumeration value="4"/>
<xs:enumeration value="5"/>
<xs:enumeration value="6"/>
<xs:enumeration value="7"/>

</xs:restriction>

</xs:simpleType>

</xs:schema>
```

### 3. Feladat második része

#### 3.1. Adatbázis beolvasása DOM segítségével

Az XML dokumentumból felolvasok minden elementet azok adatait kiírom a konzolra.

```
package hu.domparse.wuwios;
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 DomReadwuwios {
    public static void main(String[] args){
        NodeList list;
        NodeList list2;
        try {
            DocumentBuilderFactory
factory=DocumentBuilderFactory.newInstance();
            DocumentBuilder documentBuilder=factory.newDocumentBuilder();
```



```

        //fájl beolvasása
        Document document=documentBuilder.parse(new
File("XMLWUWIOS.xml"));
        document.getDocumentElement().normalize();
        //root elem megkeresése
        System.out.println("Root element : " +
document.getDocumentElement().getNodeName());
        System.out.println("~~~~~");
        //Aktuális elem meghatározása
        list=document.getElementsByTagName("videoteka");
        for (int i=0;i<list.getLength();i++) {
            Node node=list.item(i);
            System.out.println("\nAktuális elem: " +
node.getNodeName());

            //Videótéka adatainak kiírása
            if(node.getNodeType() == Node.ELEMENT_NODE) {
                Element element=(Element) node;
                System.out.println("Videótéka id: " +
element.getAttribute("tk_id"));
                System.out.println("Neve: "
+
element.getElementsByTagName("nev").item(0).getTextContent());
                //Videótéka címének kiírása (összetett elem)
                Node nodecimt=list.item(i);
                if(nodecimt.getNodeType()==Node.ELEMENT_NODE){
                    Element cimelement=(Element) node;
                    System.out.println("Cím: " +
cimelement.getElementsByTagName("iranyitosz").item(0).getTextContent() + ",
" + cimelement.getElementsByTagName("varos").item(0).getTextContent() + ", "
+ cimelement.getElementsByTagName("utcahsz").item(0).getTextContent());
                }
            }
        }

        list=document.getElementsByTagName("dolgozo");
        for (int i=0;i<list.getLength();i++) {
            Node node=list.item(i);
            System.out.println("\nAktuális elem: " +
node.getNodeName());

            //Dolgozó adatainak kiírása
            if(node.getNodeType() == Node.ELEMENT_NODE) {

```

```

        Element element=(Element) node;
        System.out.println("Dolgozó id: " +
element.getAttribute("d_id"));
        System.out.println("Videótéka id: " +
element.getAttribute("tk_id"));
        System.out.println("Neve: "
+
element.getElementsByTagName("nev").item(0).getTextContent());
        System.out.println("Beosztása: "
+
element.getElementsByTagName("beosztas").item(0).getTextContent());
        System.out.println("Bére: "
+
element.getElementsByTagName("ber").item(0).getTextContent());
    }
}
list=document.getElementsByTagName("videoteka");

for (int i=0;i<list.getLength();i++) {
    Node node=list.item(i);
    //Videótéka kiválasztása, hogy a hozzá tartozó tételeket
adatait kiírtam
    if(node.getNodeType() == Node.ELEMENT_NODE) {
        Element element=(Element) node;
        System.out.println("~~~~~");
        System.out.println("\n" +
element.getElementsByTagName("nev").item(0).getTextContent()+ " tételei:");
    }
    list2=document.getElementsByTagName("tetel");
    for (int j=0;j<list2.getLength();j++) {
        Node node2=list2.item(j);
        //Tételek kiírása melyek a megfelelő tékához tartoznak
        if(node2.getNodeType() == Node.ELEMENT_NODE) {
            Element element2=(Element) node2;
            if(
Integer.parseInt(element2.getAttributees().getNamedItem("tk_id").getNodeValue
()) == i+1){

                System.out.println("\n" +
element2.getElementsByTagName("cim").item(0).getTextContent());

```

```

        System.out.println("Tétel id: " +
element2.getAttribute("t_id"));
        System.out.println("Típus: "
+
element2.getElementsByTagName("tipus").item(0).getTextContent());
        System.out.println("Megjelenés: "
+
element2.getElementsByTagName("megjelenes").item(0).getTextContent());
        System.out.println("Ár: "
+
element2.getElementsByTagName("ar").item(0).getTextContent());
        System.out.println("Kor határ: "
+
element2.getElementsByTagName("korhatar").item(0).getTextContent());
    }
}
}

list=document.getElementsByTagName("kuncsakft");
for (int i=0;i<list.getLength();i++) {
    Node node=list.item(i);
    System.out.println("\nAktuális elem: " +
node.getNodeName());
    //Kuncsakft adatainak kiírása
    if(node.getNodeType() == Node.ELEMENT_NODE) {
        Element element=(Element) node;
        System.out.println("Kuncsakft id: " +
element.getAttribute("k_id"));
        System.out.println("Videótéka id: " +
element.getAttribute("tk_id"));
        System.out.println("Neve: "
+
element.getElementsByTagName("nev").item(0).getTextContent());
        System.out.println("Születési dátum: "
+
element.getElementsByTagName("szuld").item(0).getTextContent());

        //Videótéka címének kiírása (összetett elem)
        Node nodecimt=list.item(i);
        if(nodecimt.getNodeType()==Node.ELEMENT_NODE){

```

```

        Element cimelement=(Element) node;
        System.out.println("Cím: " +
cimelement.getElementsByTagName("iranyitosz").item(0).getTextContent() + ",
" + cimelement.getElementsByTagName("varos").item(0).getTextContent() + ", "
+ cimelement.getElementsByTagName("utcahsz").item(0).getTextContent());
    }
}

}

list=document.getElementsByTagName("kolcsonzes");
for (int i=0;i<list.getLength();i++) {
    Node node=list.item(i);
    System.out.println("\nAktuális elem: " +
node.getNodeName());
    //Dolgozó adatainak kiírása
    if(node.getNodeType() == Node.ELEMENT_NODE) {
        Element element=(Element) node;
        System.out.println("Kölcsönzés id: " +
element.getAttribute("kk_id"));
        Node nodetetelek=list.item(i);
        if(nodetetelek.getNodeType()==Node.ELEMENT_NODE){
            Element tetelelement=(Element) node;
            for (int
j=0;j<tetelelement.getElementsByTagName("t").getLength();j++) {
                System.out.println("Kölcsönzések : "
+
tetelelement.getElementsByTagName("t").item(j).getTextContent());
            }
        }
    }
}

list=document.getElementsByTagName("k_kk");
for (int i=0;i<list.getLength();i++) {
    Node node=list.item(i);
    System.out.println("\nAktuális elem: kuncsaft-kölcsönzés
kapcsolat");
    //Kuncsaft adatainak kiírása

```

```

        if(node.getNodeType() == Node.ELEMENT_NODE) {
            Element element=(Element) node;
            System.out.println("Kuncsaft id: " +
element.getAttribute("k_id"));
            System.out.println("Kolcsonzes id: " +
element.getAttribute("kk_id"));
            System.out.println("Kolcsonzes kezdete: "
+
element.getElementsByTagName("idopont").item(0).getTextContent());
        }

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

```

### 3.2. Adatbázis módosítása DOM segítségével

A következő módosításokat hajtom végre: videó téka nevének átírása “Menőbb TK”-ra, a Kazetta tételek átírása DVD tételre, 1600nál kisebb árú tételek átírása 1600-as árra, a tulajdonos adószámának átírása, és a Kiss Virág nevű kuncsaft születési dátumának átírása.

```

package hu.domparse.wuwios;

import org.w3c.dom.Document;

import org.w3c.dom.Element;

import org.w3c.dom.Node;

import org.w3c.dom.NodeList;

import javax.xml.parsers.DocumentBuilder;

import javax.xml.parsers.DocumentBuilderFactory;

import javax.xml.parsers.ParserConfigurationException;

import javax.xml.transform.Result;

import javax.xml.transform.Source;

```

```
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;
import java.io.File;

public class DOMModifywuwios {

    public static void main(String[] args) {

        try {

            NodeList nodeList;

            DocumentBuilderFactory factory
=DocumentBuilderFactory.newInstance();

            DocumentBuilder builder=factory.newDocumentBuilder();

            //Fájl beolvasása

            Document document=builder.parse(new File("XMLWUWIOS.xml"));

            document.getDocumentElement().normalize();

            //videóteca nevének átírása

            //Aktuális elem meghatározása

            nodeList=document.getElementsByTagName("videoteka");

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

                Node node= nodeList.item(i);

                //Videóteca nevének átírása.

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

                    Element element=(Element) node;

                    element.getElementsByTagName("nev").item(0).setTextContent("Menőbb TK");

                }

            }

            //Minden kazetta átírása DVDre

            //Aktuális elem meghatározása

            nodeList=document.getElementsByTagName("tete1");
```

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

    Node node= nodeList.item(i);

    //tétel átírása

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

        Element element=(Element) node;

        if(element.getElementsByTagName("tipus").item(0).getTextContent().equals("Ka
zetta") )

            {

                element.getElementsByTagName("tipus").item(0).setTextContent("DVD");

            }

        }

    }

    //árak egységesítése

    nodeList=document.getElementsByTagName("tétel");

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

        Node node= nodeList.item(i);

        //tétel átírása

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

            Element element=(Element) node;

            if(Integer.parseInt(element.getElementsByTagName("ar").item(0).getTextConten
t()) < 1600)

                {

                    element.getElementsByTagName("ar").item(0).setTextContent("1600");

                }

            }

        }

    }

    //tulajdonosadószámok átírása
```

```
//Aktuális elem meghatározása
nodeList=document.getElementsByTagName("tulajdonos");
for(int i =0;i<nodeList.getLength();i++){
    Node node= nodeList.item(i);
    if(node.getNodeType()==Node.ELEMENT_NODE){
        Element element=(Element) node;

element.getElementsByTagName("adoszam").item(0).setTextContent("321");
    }
}

//Kiss Virág születési dátumának átírása
//Aktuális elem meghatározása
nodeList=document.getElementsByTagName("kunceaft");
for(int i =0;i<nodeList.getLength();i++){
    Node node= nodeList.item(i);
    //tétel átírása
    if(node.getNodeType()==Node.ELEMENT_NODE){
        Element element=(Element) node;

if(element.getElementsByTagName("nev").item(0).getTextContent().equals("Kiss
Virág") )
    {

element.getElementsByTagName("szuld").item(0).setTextContent("2022.05.07.");
    }
}

}

//Módosított xml fájl létrehozása
Transformer transformer=
TransformerFactory.newInstance().newTransformer();
Source input=new DOMSource(document);
```



```
        Result output=new StreamResult(new File("XMLWUWIOSModify.xml"));

        transformer.transform(input,output);
    }catch (ParserConfigurationException e){

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

        e.printStackTrace();
    }
}

}
```

### 3.3. Adatbázis lekérdezése DOM segítségével

A következő lekérdezéseket hajtom végre: 20évnél idősebb kuncsaftok listázása, 1600-nál olcsóbb tétetek listázása, Nagy Endre nevű kuncsaft kölcsönzéseinek listázása. A tétetek felsorolása videótéka szerint. 16 éven felülieknek ajánlott tétetek listázása, illetve Kazetta típusu tétetek listázása.

```
package hu.domparsing.wuwios;

import java.io.File;
import java.io.IOException;
import java.time.*;
import java.time.format.DateTimeFormatter;
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 DOMQuerywuwios {

    public static void main(String[] args) {

        try {

            DateTimeFormatter formatter =
                DateTimeFormatter.ofPattern("yyyy.MM.dd.");

            LocalDate most = LocalDate.now();

            NodeList nodeList;

            NodeList nodeList2;

            DocumentBuilderFactory factory
            =DocumentBuilderFactory.newInstance();

            DocumentBuilder builder=factory.newDocumentBuilder();
```

```

//Fájl beolvasása
Document document=builder.parse(new File("XMLWUWIOS.xml"));
document.getDocumentElement().normalize();

//20 évnél idősebb kuncsaftok adatainak kiírása
//Aktuális elem meghatározása
nodeList=document.getElementsByTagName("kuncsaft");
System.out.println("20évnél idősebb kuncsaftok kiírása:
\n~~~~~");

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

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

        String szuld=
element.getElementsByTagName("szuld").item(0).getTextContent();
        if(Period.between(LocalDate.parse(szuld,formatter),
most).getYears(>20)
        {
            System.out.println("\nAktuális elem: " +
node.getNodeName());

            System.out.println("~~~~~");

            System.out.println("Neve: " +
element.getElementsByTagName("nev").item(0).getTextContent());

            System.out.println("Születési dátuma: " +
element.getElementsByTagName("szuld").item(0).getTextContent());

            System.out.println("Életkora: "+
Period.between(LocalDate.parse(szuld,formatter), most).getYears());

            Node nodecimt=nodeList.item(i);

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

```

```

        System.out.println("Lakcíme: " +
cimelement.getElementsByTagName("iranyitosz").item(0).getTextContent() + ",
" + cimelement.getElementsByTagName("varos").item(0).getTextContent() + ", "
+ cimelement.getElementsByTagName("utcahsz").item(0).getTextContent());

    }

}

}

}

//Nagy Endre nevű kuncsaft kölcsönzéseinek kiírása
nodeList=document.getElementsByTagName("kuncsaft");

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

    Node node = nodeList.item(i);

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

        Element element=(Element) node;

        String nev =
element.getElementsByTagName("nev").item(0).getTextContent();

        if(nev.equals("Nagy Endre"))

        {

            int ne_id
=Integer.parseInt(element.getAttributes().getNamedItem("k_id").getNodeValue(
));

            nodeList2=document.getElementsByTagName("k_kk");

            for (int j = 0; j < nodeList2.getLength(); j++)

            {

                Node node2=nodeList2.item(i);

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

                    Element element2=(Element) node2;

                    System.out.println("~");

```

```

if(Integer.parseInt(element2.getAttributes().getNamedItem("k_id").getNodeValue()) == ne_id){

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

    System.out.println("~~~~~");

    System.out.println("Időpont: " + element2.getElementsByTagName("idopont").item(0).getTextContent());

}

}

}

}

}

}

//1600-nál olcsóbb tételek listázása
//Aktuális elem meghatározása
nodeList=document.getElementsByTagName("tetel");

System.out.println("\n~~~~~ \n1600-nál olcsóbb tételek listázása: \n~~~~~");

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

    Node node= nodeList.item(i);

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

        Element element=(Element) node;

        if(Integer.parseInt(element.getElementsByTagName("ar").item(0).getTextContent()) < 1600)

            {

```

```

        System.out.println("\n" +
element.getElementsByTagName("cim").item(0).getTextContent());

        System.out.println("Tétel id: " +
element.getAttribute("t_id"));

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

        System.out.println("Megjelenés: "
+
element.getElementsByTagName("megjelenes").item(0).getTextContent());

        System.out.println("Ár: "
+
element.getElementsByTagName("ar").item(0).getTextContent());

        System.out.println("Kor határ: "
+
element.getElementsByTagName("korhatar").item(0).getTextContent());
    }
}

//tételek felsorolása videó téka szerint
nodeList=document.getElementsByTagName("videoteka");

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

//Videótéka kiválasztása, hogy a hozzá tartozó tételeket
adatait kiírjam

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

    Element element=(Element) node;

    System.out.println("~~~~~");

    System.out.println("\n" +
element.getElementsByTagName("nev").item(0).getTextContent()+ " tételei:");

}
}

```

```

nodeList2=document.getElementsByTagName("tetel");

for (int j=0;j<nodeList2.getLength();j++) {

    Node node2=nodeList2.item(j);

    //Tételek kiírása melyek a megfelelő tékához tartoznak

    if(node2.getNodeType() == Node.ELEMENT_NODE) {

        Element element2=(Element) node2;

        if(
Integer.parseInt(element2.getAttribute().getNamedItem("tk_id").getNodeValue
()) == i+1){

            System.out.println("\n" +
element2.getElementsByTagName("cim").item(0).getTextContent());

            System.out.println("Tétel id: " +
element2.getAttribute("t_id"));

            System.out.println("Típus: "
+
element2.getElementsByTagName("tipus").item(0).getTextContent());

            System.out.println("Megjelenés: "
+
element2.getElementsByTagName("megjelenes").item(0).getTextContent());

            System.out.println("Ár: "
+
element2.getElementsByTagName("ar").item(0).getTextContent());

            System.out.println("Kor határ: "
+
element2.getElementsByTagName("korhatar").item(0).getTextContent());

        }

    }

}

//16 éven felülieknek ajánlott filmek

//Aktuális elem meghatározása

```

```

        nodeList=document.getElementsByTagName("tetel");

        System.out.println("\n~~~~~ \n16 éven felülieknek:
\n~~~~~");

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

            Node node= nodeList.item(i);

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

                Element element=(Element) node;

                if(Integer.parseInt(element.getElementsByTagName("korhatar").item(0).getText
                Content()) == 16)

                    {

                        System.out.println("\n" +
                        element.getElementsByTagName("cim").item(0).getTextContent());

                        System.out.println("Tétel id: " +
                        element.getAttribute("t_id"));

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

                        System.out.println("Megjelenés: "
                        +
                        element.getElementsByTagName("megjelenes").item(0).getTextContent());

                        System.out.println("Ár: "
                        +
                        element.getElementsByTagName("ar").item(0).getTextContent());

                        System.out.println("Kor határ: "
                        +
                        element.getElementsByTagName("korhatar").item(0).getTextContent());

                    }

                }

            }

        }

        //Kazetta típusú tételek listázása
    
```



```

        nodeList=document.getElementsByTagName("tetel");

        System.out.println("\n~~~~~ \nKazetták listázása:
\n~~~~~");

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

            Node node= nodeList.item(i);

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

                Element element=(Element) node;

                if(element.getElementsByTagName("tipus").item(0).getTextContent().equals("Ka
                zetta") )

                    {

                        System.out.println("\n" +
                        element.getElementsByTagName("cim").item(0).getTextContent());

                        System.out.println("Tétel id: " +
                        element.getAttribute("t_id"));

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

                        System.out.println("Megjelenés: "
                        +
                        element.getElementsByTagName("megjelenes").item(0).getTextContent());

                        System.out.println("Ár: "
                        +
                        element.getElementsByTagName("ar").item(0).getTextContent());

                        System.out.println("Kor határ: "
                        +
                        element.getElementsByTagName("korhatar").item(0).getTextContent());

                    }

                }

            }

        }catch (ParserConfigurationException | IOException | SAXException
        e){

```

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