

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Készítette: **Szarka Péter**

Neptunkód: **NQIXT9**

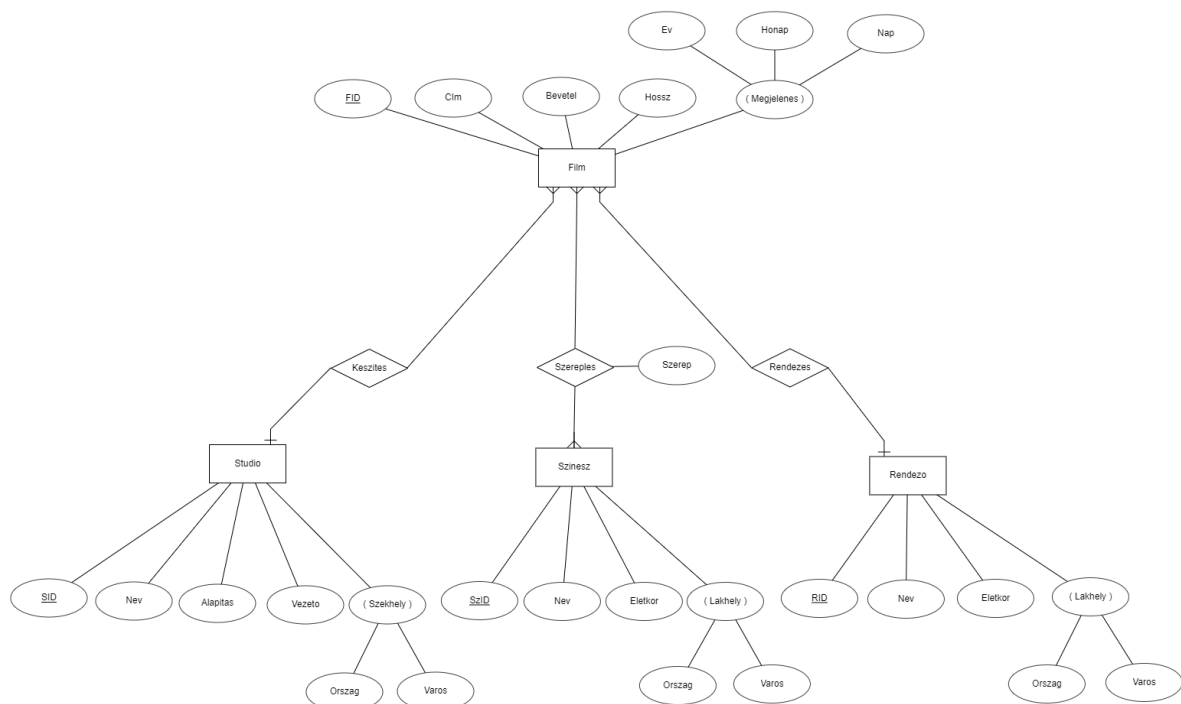
Tartalomjegyzék

1. feladat	3
1a) Az adatbázis ER modellje	3
1b) Az adatbázis konvertálása XDM modellre	4
1c) Az XDM modell alapján XML dokumentum készítése	5
1d) Az XML dokumentum alapján XMLSchema készítése	9
2. feladat	12
2a) Adatolvasás	12
2b) Adatmódosítás	16
3c) Adatlekérdezés	21

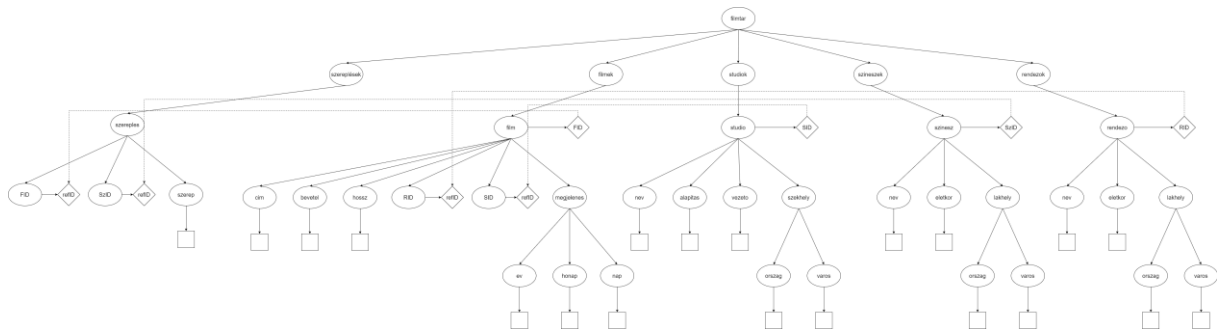
1. feladat

A választott beadandó feladatom filmek nyilvántartásához való xml dokumentum elkészítése. Nem csupán a filmek, de az ahhoz tartozó stúdiók és rendezők illetve persze a benne játszó színészek is belekerültek a dokumentumba. A filmekről a címük és a megjelenés pontos dátuma mellett eltárolásra kerül még a terjedelmek és a bevételük. A stúdiókról eltároljuk a nevüket, az alapításuk évét, a vezérigazgatójának a nevét és az országot és várost ahol a központja található ennek az egésznek. A színészekről csak néhány adatot tarolunk, amelyek a nevük, az életkoruk és legutoljára a lakhelyük országra és városra pontosan. A rendezőkről hasonlóan a színészekhez nem tárolunk túl sok információt csak, mint a színészeknél a nevüket, a korukat, valamelyest a tartózkodásuknak a helyét. Ezeken kívül még rendelkeznek ID-val is, amely a pontos azonosításért fellel, hiszen több filmből is készült újrafeldolgozás, amelyek ugyan azt a címet viselik, mint az előző. Illetve annak sincs kizárva a lehetősége, hogy két vagy több ugyanolyan nevű színész vagy rendező legyen. Illetve nem egy filmbe nem egy szereplő van, így azokat az ID segítségével könnyen rendszerezhető és még a karakter nevét is el tudjuk menteni melléjük.

1a) Az adatbázis ER modellje



1b) Az adatbázis konvertálása XDM modellre



1c) Az XDM modell alapján XML dokumentum készítése

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><?xml-model
href="XMLSchemaNQIXT9.xsd" type="application/xml"
schematypens="http://www.w3.org/2001/XMLSchema"?><filmtar>
  <rendezok>
    <rendezo RID="r01">
      <nev>David Zucker</nev>
      <eletkor>74</eletkor>
      <lakhely>
        <orszag>USA</orszag>
        <varos>Los Angeles</varos>
      </lakhely>
    </rendezo>
    <rendezo RID="r02">
      <nev>Peter Segal</nev>
      <eletkor>58</eletkor>
      <lakhely>
        <orszag>USA</orszag>
        <varos>Chicago</varos>
      </lakhely>
    </rendezo>
    <rendezo RID="r03">
      <nev>Keenen Ivory Wayans</nev>
      <eletkor>63</eletkor>
      <lakhely>
        <orszag>USA</orszag>
        <varos>New York</varos>
      </lakhely>
    </rendezo>
  </rendezok>
  <szineszek>
    <szinesz SzID="sz01">
      <nev>Leslie Nielsen</nev>
      <eletkor>84</eletkor>
      <lakhely>
        <orszag>USA</orszag>
        <varos>Fort Lauderdale</varos>
      </lakhely>
    </szinesz>
    <szinesz SzID="sz02">
      <nev>Priscilla Presley</nev>
      <eletkor>76</eletkor>
      <lakhely>
        <orszag>USA</orszag>
        <varos>Los Angeles</varos>
      </lakhely>
    </szinesz>
    <szinesz SzID="sz03">
```

```

        <nev>Marlon Wayans</nev>
        <etekor>49</etekor>
        <lakhely>
            <orszag>USA</orszag>
            <varos>Los Angeles</varos>
        </lakhely>
    </szinesz>
</szineszek>
<studiok>
    <studio SID="s01">
        <nev>Paramount Pictures</nev>
        <alapitas>1912</alapitas>
        <vezeto>Adolf Zukor</vezeto>
        <szekhely>
            <orszag>USA</orszag>
            <varos>Los Angeles</varos>
        </szekhely>
    </studio>
    <studio SID="s02">
        <nev>Miramax</nev>
        <alapitas>1979</alapitas>
        <vezeto>Bob Weinstein</vezeto>
        <szekhely>
            <orszag>USA</orszag>
            <varos>Santa Monica</varos>
        </szekhely>
    </studio>
</studiok>
<filmek>
    <film FID="f01">
        <cim>Csupasz pisztoly</cim>
        <bevetel>78756177</bevetel>
        <hossz>85</hossz>
        <megjelenes>
            <ev>1988</ev>
            <honap>12</honap>
            <nap>2</nap>
        </megjelenes>
        <RID refID="r01"/>
        <SID refID="s01"/>
    </film>
    <film FID="f02">
        <cim>Csupasz pisztoly 2 és 1/2</cim>
        <bevetel>192000000</bevetel>
        <hossz>81</hossz>
        <megjelenes>
            <ev>1991</ev>
            <honap>6</honap>
            <nap>28</nap>
        </megjelenes>
    </film>
</filmek>

```

```

        </megjelenes>
        <RID refID="r01"/>
        <SID refID="s01"/>
    </film>
    <film FID="f03">
        <cim>Csúszás pisztoly 33 1/3 – Az utolsó merénylet</cim>
        <bevetel>51132598</bevetel>
        <hossz>81</hossz>
        <megjelenes>
            <ev>1994</ev>
            <honap>3</honap>
            <nap>18</nap>
        </megjelenes>
        <RID refID="r02"/>
        <SID refID="s01"/>
    </film>
    <film FID="f04">
        <cim>Horrorra akadva 2</cim>
        <bevetel>71308997</bevetel>
        <hossz>83</hossz>
        <megjelenes>
            <ev>2001</ev>
            <honap>7</honap>
            <nap>4</nap>
        </megjelenes>
        <RID refID="r03"/>
        <SID refID="s02"/>
    </film>
</filmek>
<szereplesek>
    <szereples>
        <FID refID="f01"/>
        <SzID refID="sz01"/>
        <szerep>Frank Drebin hadnagy</szerep>
    </szereples>
    <szereples>
        <FID refID="f01"/>
        <SzID refID="sz02"/>
        <szerep>Jane Spencer</szerep>
    </szereples>
    <szereples>
        <FID refID="f02"/>
        <SzID refID="sz01"/>
        <szerep>Frank Drebin hadnagy</szerep>
    </szereples>
    <szereples>
        <FID refID="f02"/>
        <SzID refID="sz02"/>
        <szerep>Jane Spencer</szerep>
    </szereples>

```

```
</szereples>
<szereples>
  <FID refID="f03"/>
  <SzID refID="sz01"/>
  <szerep>Frank Drebin hadnagy</szerep>
</szereples>
<szereples>
  <FID refID="f03"/>
  <SzID refID="sz02"/>
  <szerep>Jane Spencer</szerep>
</szereples>
<szereples>
  <FID refID="f04"/>
  <SzID refID="sz03"/>
  <szerep>Kurta Meeks</szerep>
</szereples>
</szereplesek>
</filmtar>
```


1d) Az XML dokumentum alapján XMLSchema készítése

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:element name="filmtar">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="rendezok" type="rendezotype" maxOccurs="1"/>
        <xs:element name="szineszek" type="szinesztype"
maxOccurs="1"/>
        <xs:element name="studiok" type="studiotype" maxOccurs="1"/>
        <xs:element name="filmek" type="filmtype" maxOccurs="1"/>
        <xs:element name="szereplesek" type="szereplestype"
maxOccurs="1"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:complexType name="rendezotype">
    <xs:sequence>
      <xs:element name="rendezo" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="nev" type="xs:string"/>
            <xs:element name="eletkor" type="xs:integer"/>
            <xs:element name="lakhely" type="lakszekhelytype"/>
          </xs:sequence>
          <xs:attribute name="RID" type="xs:ID" use="required"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="megjelenestype">
    <xs:sequence>
      <xs:element name="ev" type="xs:integer"/>
      <xs:element name="honap" type="xs:integer"/>
      <xs:element name="nap" type="xs:integer"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="lakszekhelytype">
    <xs:sequence>
      <xs:element name="orszag" type="xs:string"/>
      <xs:element name="varos" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

```

</xs:complexType>

<xs:complexType name="szinesztype">
  <xs:sequence>
    <xs:element name="szinesz" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="nev" type="xs:string"/>
          <xs:element name="eletkor" type="xs:integer"/>
          <xs:element name="lakhely" type="lakszekhelytype"/>
        </xs:sequence>
        <xs:attribute name="SzID" type="xs:ID" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="studiotype">
  <xs:sequence>
    <xs:element name="studio" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="nev" type="xs:string"/>
          <xs:element name="alapitas" type="xs:integer"/>
          <xs:element name="vezeto" type="xs:string"/>
          <xs:element name="szekhely" type="lakszekhelytype"/>
        </xs:sequence>
        <xs:attribute name="SID" type="xs:ID" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="filmtype">
  <xs:sequence>
    <xs:element name="film" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="cim" type="xs:string"/>
          <xs:element name="bevetel" type="xs:long"/>
          <xs:element name="hossz" type="xs:integer"/>
          <xs:element name="megjelenes" type="megjelenestype"/>
          <xs:element name="RID">
            <xs:complexType>
              <xs:attribute name="refID" type="xs:IDREF"/>
            </xs:complexType>
          </xs:element>
          <xs:element name="SID">

```

```

        <xs:complexType>
            <xs:attribute name="refID" type="xs:IDREF"/>
        </xs:complexType>
    </xs:element>
</xs:sequence>
    <xs:attribute name="FID" type="xs:ID" use="required"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>

<xs:complexType name="szereplestype">
    <xs:sequence>
        <xs:element name="szereples" maxOccurs="unbounded">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="FID">
                        <xs:complexType>
                            <xs:attribute name="refID" type="xs:IDREF"/>
                        </xs:complexType>
                    </xs:element>
                    <xs:element name="SzID">
                        <xs:complexType>
                            <xs:attribute name="refID" type="xs:IDREF"/>
                        </xs:complexType>
                    </xs:element>
                    <xs:element name="szerep" type="xs:string"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>
</xs:sequence>
</xs:complexType>

</xs:schema>

```

2. feladat

2a) Adatolvasás

A kód:

```
package hu.domparsa.nqixt9;

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

public class DOMReadNQIXT9 {

    public static void main(String[] args) {
        // A DOM létrehozása az XML dokumentumból
        DocumentBuilderFactory documentBuilderFactory =
DocumentBuilderFactory.newInstance();
        try {
            DocumentBuilder documentBuilder =
documentBuilderFactory.newDocumentBuilder();
            Document document = documentBuilder.parse(new
File("XMLNQIXT9.xml"));
            document.getDocumentElement().normalize();//normalizálás
            Node root = document.getDocumentElement();//root elem
kiválasztása
            listAll(root);
        } catch (ParserConfigurationException | SAXException | IOException e)
{
            e.printStackTrace();
        }
    }

    public static void listAll(Node root) {
        //root elem gyerek elemeinek kiválasztása
        NodeList nodeList = root.getChildNodes();
        //kiíratáshoz String-ek
        String rendezo = "Rendezok\n\n";
        String szinesz = "Szineszek\n\n";
        String studio = "Studiok\n\n";
        String film = "Filmek\n\n";
        String szereples = "Szereplesek\n\n";
    }
}
```

```

        for (int i = 0; i < nodeList.getLength(); i++) {
            Node node = nodeList.item(i); //root gyerek elemei egyessével
            if (node.getNodeType() == Node.ELEMENT_NODE) {
                //root gyerek elemeinek a gyerekelemeinek a kiválasztása
                NodeList subNodeList = nodeList.item(i).getChildNodes();
                for (int j = 0; j < subNodeList.getLength(); j++) {
                    //root gyerek elemeinek a gyerekelemei egyessével
                    Node subNode = subNodeList.item(j);
                    if (subNode.getNodeType() == Node.ELEMENT_NODE) {
                        //Stringekbe csoportosítás
                        rendezo += oneNode(subNode, "rendezo");
                        szinesz += oneNode(subNode, "szinesz");
                        studio += oneNode(subNode, "studio");
                        film += oneNode(subNode, "film");
                        szereples += oneNode(subNode, "szereples");
                    }
                }
            }
        }
        //kiiratások
        System.out.println(rendezo);
        System.out.println(szinesz);
        System.out.println(studio);
        System.out.println(film);
        System.out.println(szereples);
    }

    public static String oneNode(Node subNode, String tag) {
        String out = "";
        if (subNode.getNodeName().equals(tag)) { //tag String-gel megegyezik
            nev node-ot keres
            //ha van attribute-ja akkor kiírja a legelsőt a String-be
            if (subNode.getAttributes().getLength() > 0)
            {
                out += tag + " ID : " +
                subNode.getAttributes().item(0).getTextContent() + "\n";
            }
            NodeList subSubNodeList = subNode.getChildNodes();
            for (int k = 0; k < subSubNodeList.getLength(); k++) {
                Node subSubNode = subSubNodeList.item(k); //node gyerek elemei
                egyessével
                if (subSubNode.getNodeType() == Node.ELEMENT_NODE) {
                    //az elem nevét és főtípusát megadja a String-be
                    switch (subSubNode.getNodeName()) {
                        //a refID-kat tartalmazó elemeket olvasd ki
                        case "RID":
                        case "SzID":
                        case "SID":

```

```

        case "FID":
            //az els attribute-omokat rakja bele a String-be
            out +=subSubNode.getNodeName() + " : "
            +
subSubNode.getAttributes().item(0).getTextContent()+"\n";
            break;
            case "megjelenes":
            case "lakhely":
            case "szekhely":
                NodeList sssNode = subSubNode.getChildNodes();
                //a gyerekelemeinket ciklussal végigmegyünk és meggy
rakja bele a String-be
                for (int l = 0; l < sssNode.getLength(); l++) {
                    if (sssNode.item(l).getNodeType() ==
Node.ELEMENT_NODE) {
                        out += subSubNode.getNodeName() + "- "
                        + sssNode.item(l).getNodeName() + " : " +
sssNode.item(l).getTextContent()+"\n";
                    }
                }
            break;
            default:
                //normál esetben az elem nevét és a kontextusát
rakja bele a String-be
                out += subSubNode.getNodeName() + " : " +
subSubNode.getTextContent()+"\n";
            }
        }
        out+="\n";
    }
    return out;
}
}

```

A kimenete:

```
1 Rendezők
2
3 rendező ID : r01
4 nev : David Zucker
5 elatkor : 34
6 lakhely-oroszag : USA
7 lakhely-varos : Los Angeles
8
9 rendező ID : r02
10 nev : Peter Segal
11 elatkor : 58
12 lakhely-oroszag : USA
13 lakhely-varos : Chicago
14
15 rendező ID : r03
16 nev : Keenan Ivory Noyans
17 elatkor : 63
18 lakhely-oroszag : USA
19 lakhely-varos : New York
20
21
22 Színeszek
23
24 színész ID : sz01
25 nev : Leslie Nielsen
26 elatkor : 61
27 lakhely-oroszag : USA
28 lakhely-varos : Fort Lauderdale
29
30 színész ID : sz02
31 nev : Patricia Presley
32 elatkor : 76
33 lakhely-oroszag : USA
34 lakhely-varos : Los Angeles
35
36 színész ID : sz03
37 nev : Marlon Mayans
38 elatkor : 49
39 lakhely-oroszag : USA
40 lakhely-varos : Los Angeles
41
42
43 Színház
44
45 studio ID : s01
46 nev : Paramount Pictures
47 alapítás : 1912
48 vezető : Adolf Zukor
49 színhely-oroszag : USA
50 színhely-varos : Los Angeles
51
52 studio ID : s02
53 nev : Miramax
54 alapítás : 1979
55 vezető : Bob Weinstein
56 színhely-oroszag : USA
57 színhely-varos : Santa Monica
58
59
60 Filmek
61
62 film ID : f01
63 cím : Caspaz piratraly
64 bevétel : 78756177
65 hossz : 85
66 megjelenés-év : 1988
67 megjelenés-hónap : 12
68 megjelenés-nap : 2
69 RID : r01
70 SID : s01
71
72 film ID : f02
73 cím : Caspaz piratraly 2 és 1/2
74 bevétel : 152800000
75 hossz : 81
76 megjelenés-év : 1991
77 megjelenés-hónap : 6
78 megjelenés-nap : 28
79 RID : r01
80 SID : s01
81
82 film ID : f03
83 cím : Caspaz piratraly 33 1/3 - Az utolsó merénylet
84 bevétel : 51132596
85 hossz : 81
86 megjelenés-év : 1994
87 megjelenés-hónap : 3
88 megjelenés-nap : 18
89 RID : r02
90 SID : s01
91
92 film ID : f04
93 cím : Horvára akadva 2
94 bevétel : 71300997
95 hossz : 83
96 megjelenés-év : 2001
97 megjelenés-hónap : 7
98 megjelenés-nap : 4
99 RID : r03
100 SID : s02
101
102
103 Szereplők
104
105 FID : f01
106 SID : sz01
107 szerep : Frank Drebin hadnagy
108
109 FID : f01
110 SID : sz02
111 szerep : Jane Spencer
112
113 FID : f02
114 SID : sz01
115 szerep : Frank Drebin hadnagy
116
117 FID : f02
118 SID : sz02
119 szerep : Jane Spencer
120
121 FID : f03
122 SID : sz01
123 szerep : Frank Drebin hadnagy
124
125 FID : f03
126 SID : sz02
127 szerep : Jane Spencer
128
129 FID : f04
130 SID : sz03
131 szerep : Karta Meeks
```

2b) Adatmódosítás

A kód:

```
package hu.domparsa.nqixt9;

import java.io.File;
import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.util.Scanner;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerException;
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.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;

public class DOMModifyNQIXT9 {

    public static void main(String[] args){
        // A DOM létrehozása az XML dokumentumból
        DocumentBuilderFactory documentBuilderFactory =
DocumentBuilderFactory.newInstance();
        try {
            DocumentBuilder documentBuilder =
documentBuilderFactory.newDocumentBuilder();
            Document document = documentBuilder.parse(new
File("XMLNQIXT9.xml"));
            document.getDocumentElement().normalize();//normalizálás
            Scanner scanner = new Scanner(System.in);
            boolean exit=true;
            while(exit) { //menu rendszer
                System.out.println("1 : Adj meg egy film bevetelet\n2 : Adj meg egy színész életkorát\n"
+ "3 : Adj meg egy studio új vezetőjének a nevét");
                switch(scanner.nextInt()) {
                    case 1 : filmIncome(document, scanner);break;
                    case 2 : szineszAge(document, scanner);break;
                    case 3 : studioBoss(document, scanner);break;
                    default : exit=false;
                }
            }
            scanner.close();
        }
    }
}
```



```

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

public static void filmIncome(Document document, Scanner scanner) {
    NodeList nodeList = document.getElementsByTagName("film");//film
elemek kiválasztása
    System.out.println("Adja meg a film ID-ját :");
    String input = scanner.next();//beolvasás
    for (int i = 0; i < nodeList.getLength(); i++) {
        Node node = nodeList.item(i);
        if (node.getNodeType() == Node.ELEMENT_NODE) {
            //a megfelelő ID megkeresése
            if
(node.getAttributes().getNamedItem("FID").getTextContent().equals(input)) {
                NodeList subNodeList = node.getChildNodes();
                for (int j = 0; j < subNodeList.getLength(); j++) {
                    Node subNode = subNodeList.item(j);
                    if (subNode.getNodeName().equals("bevetel"))
{//bevetel elem keresése
                        System.out.println("Adja meg a film bevétele :");
                        //új bevetel bekérése és beírása az xml-be
                        subNode.setTextContent(scanner.next());
                    }
                }
            }
        }
    }
}

public static void szineszAge(Document document, Scanner scanner) {
    NodeList nodeList = document.getElementsByTagName("szinesz");//szinesz
elemek kiválasztása
    System.out.println("Adja meg a szinesz ID-ját :");
    String input = scanner.next();//beolvasás
    for (int i = 0; i < nodeList.getLength(); i++) {
        Node node = nodeList.item(i);
        if (node.getNodeType() == Node.ELEMENT_NODE) {
            //a megfelelő ID megkeresése
            if
(node.getAttributes().getNamedItem("SzID").getTextContent().equals(input)) {
                NodeList subNodeList = node.getChildNodes();
                for (int j = 0; j < subNodeList.getLength(); j++) {
                    Node subNode = subNodeList.item(j);
                    if (subNode.getNodeName().equals("eletkor"))
{//eletkor elem keresése

```

```
//új kor bekérése és beírása az xml-be
System.out.println("Adja meg a színesz új
életkorát :");

        subNode.setTextContent(scanner.next());
    }
}
}
}
}

public static void studioBoss(Document document, Scanner scanner) {
    NodeList nodeList = document.getElementsByTagName("studio");//studio
elemek kiválasztása
    System.out.println("Adja meg a studio ID-ját :");
    String input = scanner.next();//beolvasás
    for (int i = 0; i < nodeList.getLength(); i++) {
        Node node = nodeList.item(i);
        if (node.getNodeType() == Node.ELEMENT_NODE) {
            //a megfelelő ID megkeresése
            if
(node.getAttributes().getNamedItem("SID").getTextContent().equals(input)) {
                NodeList subNodeList = node.getChildNodes();
                for (int j = 0; j < subNodeList.getLength(); j++) {
                    Node subNode = subNodeList.item(j);
                    if (subNode.getNodeName().equals("vezeto")) {//vezeto
elem kiválasztása

                        System.out.println("Adja meg a vezeto új nevét
:");

                        //új név bekérése és kiírása az xml-be
                        String name = scanner.next();
                        name +=scanner.nextLine();
                        subNode.setTextContent(name);
                    }
                }
            }
        }
    }
    //fájlba és konzolra kiírás
    public static void writeToXml(Document document) throws
TransformerException, UnsupportedEncodingException {
        TransformerFactory transformerFactory =
TransformerFactory.newInstance();
        Transformer transf = transformerFactory.newTransformer();
        DOMSource source = new DOMSource(document);
        //kimeneti csatorna konzolra
        StreamResult console = new StreamResult(System.out);
        //kimeneti csatorna fájlba
```

```
StreamResult file = new StreamResult(new File("XMLNQIXT9.out.xml"));  
// A módosítások kiírása a képernyőre  
transf.transform(source, console);  
// A módosítások kiírása új fájlba  
transf.transform(source, file);  
}  
  
}
```

A kimenet:

[illegible]

3c) Adatlekérdezés

A kód:

```
package hu.domparsa.nqixt9;

import java.io.File;
import java.io.IOException;
import java.util.Calendar;
import java.util.Scanner;

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 DOMQueryNQIXT9 {

    public static void main(String[] args){
        // A DOM létrehozása az XML dokumentumból
        DocumentBuilderFactory documentBuilderFactory =
DocumentBuilderFactory.newInstance();
        try {
            DocumentBuilder documentBuilder =
documentBuilderFactory.newDocumentBuilder();
            Document document = documentBuilder.parse(new
File("XMLNQIXT9.xml"));
            document.getDocumentElement().normalize();//normalizálás
            Scanner scanner = new Scanner(System.in);
            boolean exit=true;
            while(exit) { //menu rendszer
                System.out.println("1 : Film(ek) amik több bevettelt hoztak
mint...\n"
                                + "2 : Színész(ek) akik fiatalabbak...\n3 : Studio(k)
amik fiatalabbak mint....");
                switch(scanner.nextInt()) {
                    case 1 : filmIncome(document, scanner);break;
                    case 2 : szineszAge(document, scanner);break;
                    case 3 : studioFoundation(document, scanner);break;
                    default : exit=false;
                }
            }
            scanner.close();
        } catch (ParserConfigurationException | SAXException | IOException e)
        {
            e.printStackTrace();
        }
    }
}
```

```

    public static void filmIncome(Document document, Scanner scanner) {
        NodeList nodeList = document.getElementsByTagName("film");//film
elemek kiválasztása
        System.out.println("Adja meg mennyit :");
        Long input = scanner.nextLong();//beolvasás
        System.out.println("\n");
        for (int i = 0; i < nodeList.getLength(); i++) {
            Node node = nodeList.item(i);//film gyerek elemei egyességgel
            if (node.getNodeType() == Node.ELEMENT_NODE) {
                //root gyerek elemeinek a gyerek elemeinek a kiválasztása
                NodeList subNodeList = node.getChildNodes();
                for (int j = 0; j < subNodeList.getLength(); j++) {
                    Node subNode = subNodeList.item(j);
                    if (subNode.getNodeType() == Node.ELEMENT_NODE) {
                        if (subNode.getNodeName().equals("bevetel")) //bevetele
elem keresése
                            //a kritériumnak megfelelő elem kiválasztása
                            if (Long.parseLong(subNode.getTextContent()) > input)
                                listSub(node, "Film");
                    }
                }
            }
        }
    }

    public static void szineszAge(Document document, Scanner scanner) {
        NodeList nodeList = document.getElementsByTagName("szinesz");//szinesz
elemek kiválasztása
        System.out.println("Adja meg mennyit :");
        int input = scanner.nextInt();//beolvasás
        System.out.println("\n");
        for (int i = 0; i < nodeList.getLength(); i++) {
            Node node = nodeList.item(i);//szinesz gyerek elemei egyességgel
            if (node.getNodeType() == Node.ELEMENT_NODE) {
                //root gyerek elemeinek a gyerek elemeinek a kiválasztása
                NodeList subNodeList = node.getChildNodes();
                for (int j = 0; j < subNodeList.getLength(); j++) {
                    Node subNode = subNodeList.item(j);
                    if (subNode.getNodeType() == Node.ELEMENT_NODE) {
                        if (subNode.getNodeName().equals("eletkor")) //eletkor
elem megkeresése
                            //a kritériumnak megfelelő elem kiválasztása
                            if (Integer.parseInt(subNode.getTextContent()) <
input)
                                listSub(node, "Szinesz");
                    }
                }
            }
        }
    }

```

```

    }
}

public static void studioFoundation(Document document, Scanner scanner) {
    NodeList nodeList = document.getElementsByTagName("studio");//studio
elemek kiválasztása
    System.out.println("Adja meg mennyire 1 :");
    Long input = scanner.nextLong();//beolvasás
    System.out.println("\n");
    int year = Calendar.getInstance().get(Calendar.YEAR);
    for (int i = 0; i < nodeList.getLength(); i++) {
        Node node = nodeList.item(i);//studio gyerek elemei egyessével
        if (node.getNodeType() == Node.ELEMENT_NODE) {
            //root gyerek elemeinek a gyerek elemeinek a kiválasztása
            NodeList subNodeList = node.getChildNodes();
            for (int j = 0; j < subNodeList.getLength(); j++) {
                Node subNode = subNodeList.item(j);
                if (subNode.getNodeType() == Node.ELEMENT_NODE) {
                    if (subNode.getNodeName().equals("alapitas") )//alapitas
elem megkeresése
                        //a kritériumnak megfelelő elem kiválasztása
                        if (year - Integer.parseInt(subNode.getTextContent())
< input)
                            listSub(node, "Studio");
                }
            }
        }
    }
}

//kiírás
private static void listSub(Node node, String tag) {
    String out="";
    out +=tag + " ID : " +
node.getAttribute().item(0).getTextContent()+"\n";//ID Stringbe írása
    Element element = (Element) node;
    switch (tag) { //külön kezelése a kiírásnak
        case "Film"://film cím és bevetel String-be írása
            out +=
element.getElementsByTagName("cim").item(0).getNodeName()
            + " : "
+element.getElementsByTagName("cim").item(0).getTextContent()+"\n";
            out +=
element.getElementsByTagName("bevetel").item(0).getNodeName()
            + " : "
+element.getElementsByTagName("bevetel").item(0).getTextContent()+"\n";
            break;
        case "Szinesz"://szinesz név és életkor String-be írása
            out +=
element.getElementsByTagName("nev").item(0).getNodeName()

```

```

        + " : "
+element.getElementsByTagName("nev").item(0).getTextContent()+"\n";
        out +=
element.getElementsByTagName("eletkor").item(0).getNodeName()
        + " : "
+element.getElementsByTagName("eletkor").item(0).getTextContent()+"\n";
        break;
        case "Studio"://studio nev s alapitas String-be irsa
        out +=
element.getElementsByTagName("nev").item(0).getNodeName()
        + " : "
+element.getElementsByTagName("nev").item(0).getTextContent()+"\n";
        out +=
element.getElementsByTagName("alapitas").item(0).getNodeName()
        + " : "
+element.getElementsByTagName("alapitas").item(0).getTextContent()+"\n";
        break;
    }

    System.out.println(out);//kiir s
}
}

```


A kimenet:

```
1 1 : Film(ek) amik több bevételt hoztak mint...
2 2 : Színész(ek) akik fiatalabbak...
3 3 : Stúdió(k) amik fiatalabbak mint....
4 1
5 Adja meg mennyinél :
6 90000000
7
8
9 Film ID : f02
10 cím : Csupasz pisztoly 2 és 1/2
11 bevetel : 192000000
12
13 1 : Film(ek) amik több bevételt hoztak mint...
14 2 : Színész(ek) akik fiatalabbak...
15 3 : Stúdió(k) amik fiatalabbak mint....
16 1
17 Adja meg mennyinél :
18 9000000
19
20
21 Film ID : f01
22 cím : Csupasz pisztoly
23 bevetel : 78756177
24
25 Film ID : f02
26 cím : Csupasz pisztoly 2 és 1/2
27 bevetel : 192000000
28
29 Film ID : f03
30 cím : Csupasz pisztoly 33 1/3 - Az utolsó merénylet
31 bevetel : 51132598
32
33 Film ID : f04
34 cím : Horrorra akadva 2
35 bevetel : 71308997
36
37 1 : Film(ek) amik több bevételt hoztak mint...
38 2 : Színész(ek) akik fiatalabbak...
39 3 : Stúdió(k) amik fiatalabbak mint....
40 2
41 Adja meg mennyinél :
42 80
43
44
45 Színesz ID : sz02
46 nev : Priscilla Presley
47 életkor : 76
48
49 Színesz ID : sz03
50 nev : Marlon Wayans
51 életkor : 49
52
53 1 : Film(ek) amik több bevételt hoztak mint...
54 2 : Színész(ek) akik fiatalabbak...
55 3 : Stúdió(k) amik fiatalabbak mint....
56 3
57 Adja meg mennyinél :
58 50
59
60
61 Studio ID : s02
62 nev : Miramax
63 alapitas : 1979
64
65 1 : Film(ek) amik több bevételt hoztak mint...
66 2 : Színész(ek) akik fiatalabbak...
67 3 : Stúdió(k) amik fiatalabbak mint....
68 0
```