

# JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Készítette: **Szarka Péter**

Neptunkód: **NQIXT9**

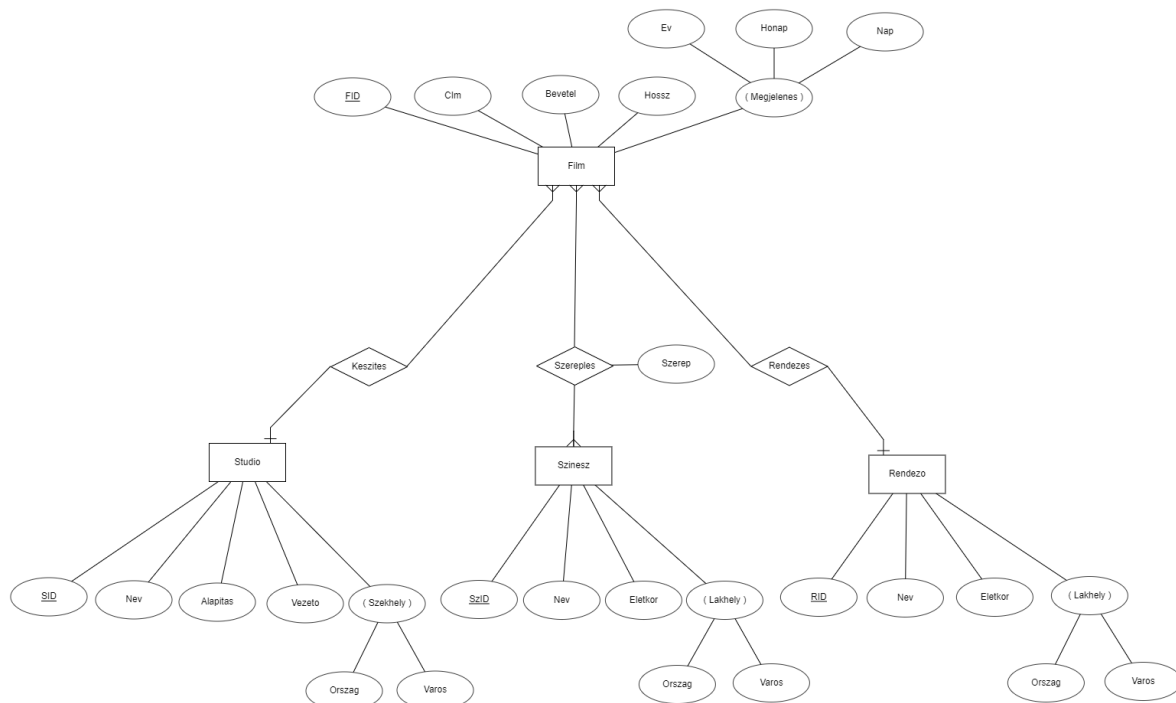
# Tartalomjegyzék

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 .....	6
2. feladat.....	7
2a) Adatolvasás .....	7
2b) Adatmódosítás.....	9
3c) Adatlekérdezés .....	11

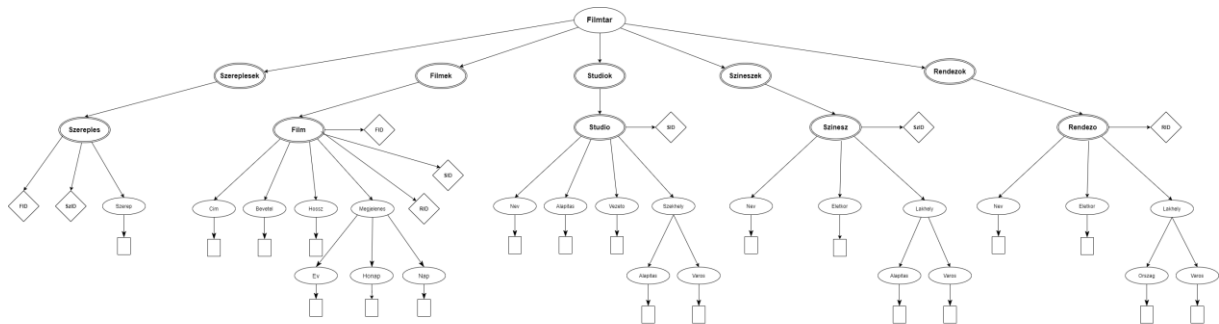
# 1. feladat

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

```
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <?xml-model href="XMLSchema.xsd" type="application/xml" schemaLocation="http://www.xml.org/2001/XMLSchema"?>
3 <filtek>
4
5     <rendezok>
6         <rendezok id="1">
7             <nev>David Zucker</nev>
8             <szuletesiId>74</szuletesiId>
9             <labaly>
10                 <orszag>USA</orszag>
11                 <varos>Los Angeles</varos>
12             </labaly>
13         </rendezok>
14         <rendezok id="2">
15             <nev>Peter Segal</nev>
16             <szuletesiId>58</szuletesiId>
17             <labaly>
18                 <orszag>USA</orszag>
19                 <varos>Chicago</varos>
20             </labaly>
21         </rendezok>
22         <rendezok id="3">
23             <nev>Keanu Reeves</nev>
24             <szuletesiId>42</szuletesiId>
25             <labaly>
26                 <orszag>USA</orszag>
27                 <varos>New York</varos>
28             </labaly>
29         </rendezok>
30         <rendezok id="4">
31             <nev>Linda Martin</nev>
32             <szuletesiId>44</szuletesiId>
33             <labaly>
34                 <orszag>USA</orszag>
35                 <varos>Fort Lauderdale</varos>
36             </labaly>
37         </rendezok>
38         <rendezok id="5">
39             <nev>Patricia Breslin</nev>
40             <szuletesiId>76</szuletesiId>
41             <labaly>
42                 <orszag>USA</orszag>
43                 <varos>Los Angeles</varos>
44             </labaly>
45         </rendezok>
46         <rendezok id="6">
47             <nev>Marlon Wayne</nev>
48             <szuletesiId>49</szuletesiId>
49             <labaly>
50                 <orszag>USA</orszag>
51                 <varos>Los Angeles</varos>
52             </labaly>
53         </rendezok>
54     </rendezok>
55     <studiok>
56         <studio id="1">
57             <nev>Paramount Pictures</nev>
58             <szuletesiId>1912</szuletesiId>
59             <varos>Adolf Zuker</varos>
60             <labaly>
61                 <orszag>USA</orszag>
62                 <varos>Los Angeles</varos>
63             </labaly>
64         </studio>
65         <studio id="2">
66             <nev>Miramax</nev>
67             <szuletesiId>1978</szuletesiId>
68             <varos>Bob Weinstein</varos>
69             <labaly>
70                 <orszag>USA</orszag>
71                 <varos>Santa Monica</varos>
72             </labaly>
73         </studio>
74     </studiok>
75     <filmek>
76         <film id="1">
77             <nev>Casper</nev>
78             <szuletesiId>7676177</szuletesiId>
79             <hossz>85</hossz>
80             <megjelentes>
81                 <id>1988</id>
82                 <hossz>12</hossz>
83                 <szag>2</szag>
84             </megjelentes>
85             <leirasiSzoveg>
86                 <id>refid="id1"/>
87                 <id>refid="id1"/>
88             </leirasiSzoveg>
89         </film>
90         <film id="2">
91             <nev>Casper</nev>
92             <szuletesiId>1000000</szuletesiId>
93             <hossz>81</hossz>
94             <megjelentes>
95                 <id>1991</id>
96                 <hossz>6</hossz>
97                 <szag>28</szag>
98             </megjelentes>
99             <leirasiSzoveg>
100                 <id>refid="id1"/>
101                 <id>refid="id1"/>
102             </leirasiSzoveg>
103         </film>
104         <film id="3">
105             <nev>Casper</nev>
106             <szuletesiId>1112598</szuletesiId>
107             <hossz>81</hossz>
108             <megjelentes>
109                 <id>1994</id>
110                 <hossz>3</hossz>
111                 <szag>18</szag>
112             </megjelentes>
113             <leirasiSzoveg>
114                 <id>refid="id1"/>
115                 <id>refid="id1"/>
116             </leirasiSzoveg>
117         </film>
118         <film id="4">
119             <nev>Marvaca</nev>
120             <szuletesiId>1338997</szuletesiId>
121             <hossz>81</hossz>
122             <megjelentes>
123                 <id>2001</id>
124                 <hossz>7</hossz>
125                 <szag>4</szag>
126             </megjelentes>
127             <leirasiSzoveg>
128                 <id>refid="id1"/>
129                 <id>refid="id1"/>
130             </leirasiSzoveg>
131         </film>
132         <film id="5">
133             <nev>Frank Drabin hadnagy</nev>
134             <szuletesiId>1338997</szuletesiId>
135             <hossz>81</hossz>
136             <megjelentes>
137                 <id>1991</id>
138                 <hossz>6</hossz>
139                 <szag>28</szag>
140             </megjelentes>
141             <leirasiSzoveg>
142                 <id>refid="id1"/>
143                 <id>refid="id1"/>
144             </leirasiSzoveg>
145         </film>
146         <film id="6">
147             <nev>Jane Spencer</nev>
148             <szuletesiId>1338997</szuletesiId>
149             <hossz>81</hossz>
150             <megjelentes>
151                 <id>1991</id>
152                 <hossz>6</hossz>
153                 <szag>28</szag>
154             </megjelentes>
155             <leirasiSzoveg>
156                 <id>refid="id1"/>
157                 <id>refid="id1"/>
158             </leirasiSzoveg>
159         </film>
160         <film id="7">
161             <nev>Frank Drabin hadnagy</nev>
162             <szuletesiId>1338997</szuletesiId>
163             <hossz>81</hossz>
164             <megjelentes>
165                 <id>1994</id>
166                 <hossz>3</hossz>
167                 <szag>18</szag>
168             </megjelentes>
169             <leirasiSzoveg>
170                 <id>refid="id1"/>
171                 <id>refid="id1"/>
172             </leirasiSzoveg>
173         </film>
174     </filmek>
175     <szereplok>
176         <szereplo id="1">
177             <nev>David Zucker</nev>
178             <szuletesiId>74</szuletesiId>
179             <labaly>
180                 <orszag>USA</orszag>
181                 <varos>Los Angeles</varos>
182             </labaly>
183         </szereplo>
184         <szereplo id="2">
185             <nev>Peter Segal</nev>
186             <szuletesiId>58</szuletesiId>
187             <labaly>
188                 <orszag>USA</orszag>
189                 <varos>Chicago</varos>
190             </labaly>
191         </szereplo>
192         <szereplo id="3">
193             <nev>Keanu Reeves</nev>
194             <szuletesiId>42</szuletesiId>
195             <labaly>
196                 <orszag>USA</orszag>
197                 <varos>New York</varos>
198             </labaly>
199         </szereplo>
200         <szereplo id="4">
201             <nev>Linda Martin</nev>
202             <szuletesiId>44</szuletesiId>
203             <labaly>
204                 <orszag>USA</orszag>
205                 <varos>Fort Lauderdale</varos>
206             </labaly>
207         </szereplo>
208         <szereplo id="5">
209             <nev>Patricia Breslin</nev>
210             <szuletesiId>76</szuletesiId>
211             <labaly>
212                 <orszag>USA</orszag>
213                 <varos>Los Angeles</varos>
214             </labaly>
215         </szereplo>
216         <szereplo id="6">
217             <nev>Marlon Wayne</nev>
218             <szuletesiId>49</szuletesiId>
219             <labaly>
220                 <orszag>USA</orszag>
221                 <varos>Los Angeles</varos>
222             </labaly>
223         </szereplo>
224     </szereplok>
225 </filtek>
```

## 1d) Az XML dokumentum alapján XMLSchema készítése

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
3   <xs:element name="filmtar">
4     <xs:complexType>
5       <xs:sequence>
6         <xs:element name="rendezok" type="rendezotYPE" maxOccurs="1"/>
7         <xs:element name="szinesiek" type="szinesitYPE" maxOccurs="1"/>
8         <xs:element name="studio" type="studiotYPE" maxOccurs="1"/>
9         <xs:element name="filmek" type="filmtYPE" maxOccurs="1"/>
10        <xs:element name="szereplesek" type="szereplestYPE" maxOccurs="1"/>
11      </xs:sequence>
12    </xs:complexType>
13  </xs:element>
14
15  <xs:complexType name="rendezotYPE">
16    <xs:sequence>
17      <xs:element name="rendezo" maxOccurs="unbounded">
18        <xs:complexType>
19          <xs:sequence>
20            <xs:element name="nev" type="xs:string"/>
21            <xs:element name="eletkor" type="xs:integer"/>
22            <xs:element name="lakhely" type="lakszekhelytYPE"/>
23          </xs:sequence>
24          <xs:attribute name="RID" type="xs:ID" use="required"/>
25        </xs:complexType>
26      </xs:element>
27    </xs:sequence>
28  </xs:complexType>
29
30  <xs:complexType name="megjelenestYPE">
31    <xs:sequence>
32      <xs:element name="ev" type="xs:integer"/>
33      <xs:element name="honap" type="xs:integer"/>
34      <xs:element name="nap" type="xs:integer"/>
35    </xs:sequence>
36  </xs:complexType>
37
38  <xs:complexType name="lakszekhelytYPE">
39    <xs:sequence>
40      <xs:element name="orszag" type="xs:string"/>
41      <xs:element name="varos" type="xs:string"/>
42    </xs:sequence>
43  </xs:complexType>
44
45  <xs:complexType name="szinesitYPE">
46    <xs:sequence>
47      <xs:element name="szinesz" maxOccurs="unbounded">
48        <xs:complexType>
49          <xs:sequence>
50            <xs:element name="nev" type="xs:string"/>
51            <xs:element name="eletkor" type="xs:integer"/>
52            <xs:element name="lakhely" type="lakszekhelytYPE"/>
53          </xs:sequence>
54          <xs:attribute name="SID" type="xs:ID" use="required"/>
55        </xs:complexType>
56      </xs:element>
57    </xs:sequence>
58  </xs:complexType>
59
60  <xs:complexType name="studiotYPE">
61    <xs:sequence>
62      <xs:element name="studio" maxOccurs="unbounded">
63        <xs:complexType>
64          <xs:sequence>
65            <xs:element name="nev" type="xs:string"/>
66            <xs:element name="alapitas" type="xs:integer"/>
67            <xs:element name="vezeto" type="xs:string"/>
68            <xs:element name="szekhely" type="lakszekhelytYPE"/>
69          </xs:sequence>
70          <xs:attribute name="SID" type="xs:ID" use="required"/>
71        </xs:complexType>
72      </xs:element>
73    </xs:sequence>
74  </xs:complexType>
75
76  <xs:complexType name="filmtYPE">
77    <xs:sequence>
78      <xs:element name="film" maxOccurs="unbounded">
79        <xs:complexType>
80          <xs:sequence>
81            <xs:element name="cin" type="xs:string"/>
82            <xs:element name="hossz" type="xs:integer"/>
83            <xs:element name="megjelenes" type="megjelenestYPE"/>
84            <xs:element name="RID" type="xs:ID" use="required"/>
85          </xs:sequence>
86          <xs:attribute name="refID" type="xs:IDREF"/>
87        </xs:complexType>
88      </xs:element>
89      <xs:element name="SID" type="xs:ID" use="required"/>
90    </xs:sequence>
91  </xs:complexType>
92
93  <xs:complexType name="szereplestYPE">
94    <xs:sequence>
95      <xs:element name="szerep" maxOccurs="unbounded">
96        <xs:complexType>
97          <xs:sequence>
98            <xs:element name="szerep" type="xs:string"/>
99          </xs:sequence>
100         <xs:attribute name="refID" type="xs:IDREF"/>
101       </xs:complexType>
102     </xs:element>
103   </xs:sequence>
104 </xs:complexType>
105
106  <xs:complexType name="szereplesek">
107    <xs:sequence>
108      <xs:element name="szerep" maxOccurs="unbounded">
109        <xs:complexType>
110          <xs:sequence>
111            <xs:element name="szerep" type="xs:string"/>
112          </xs:sequence>
113          <xs:attribute name="refID" type="xs:IDREF"/>
114        </xs:complexType>
115      </xs:element>
116    </xs:sequence>
117  </xs:complexType>
118
119  <xs:complexType name="szereplesek">
120    <xs:sequence>
121      <xs:element name="szerep" type="xs:string"/>
122    </xs:sequence>
123  </xs:complexType>
124
125 </xs:schema>
```

# 2. feladat

## 2a) Adatolvasás

A kód:

```
1 package hu.domparse.nqixt9;
2
3 import java.io.File;
4 import java.io.IOException;
5 import javax.xml.parsers.DocumentBuilder;
6 import javax.xml.parsers.DocumentBuilderFactory;
7 import javax.xml.parsers.ParserConfigurationException;
8 import org.w3c.dom.NodeList;
9 import org.xml.sax.SAXException;
10 import org.w3c.dom.Document;
11 import org.w3c.dom.Node;
12
13 public class DOMReadNQIXT9 {
14
15     public static void main(String[] args) {
16         // A DOM létrehozása az XML dokumentumból
17         DocumentBuilderFactory documentBuilderFactory = DocumentBuilderFactory.newInstance();
18         try {
19             DocumentBuilder documentBuilder = documentBuilderFactory.newDocumentBuilder();
20             Document document = documentBuilder.parse(new File("XMLNQIXT9.xml"));
21             document.getDocumentElement().normalize();
22             Node root = document.getDocumentElement(); //root elem kiválasztása
23             listAll(root);
24         } catch (ParserConfigurationException | SAXException | IOException e) {
25             e.printStackTrace();
26         }
27     }
28
29     public static void listAll(Node root) {
30         NodeList nodeList = root.getChildNodes(); //root elem gyerek elemeinek kiválasztása
31         //Kimenet: String-ek
32         String rendezo = "Rendezoek\n\n";
33         String szinesz = "Szineszek\n\n";
34         String studio = "Studiok\n\n";
35         String film = "Filmek\n\n";
36         String szereples = "Szereples\n\n";
37         for (int i = 0; i < nodeList.getLength(); i++) {
38             Node node = nodeList.item(i); //root gyerek elemek egyessével
39             if (node.getNodeType() == Node.ELEMENT_NODE) {
40                 NodeList subNodeList = nodeList.item(i).getChildNodes(); //root gyerek elemeinek a gyerek elemeinek kiválasztása
41                 for (int j = 0; j < subNodeList.getLength(); j++) {
42                     Node subNode = subNodeList.item(j); //root gyerek elemeinek a gyerek elemeinek egyessével
43                     if (subNode.getNodeType() == Node.ELEMENT_NODE) {
44                         //Kimenet: String-ek
45                         rendezo += oneNode(subNode, "rendezo");
46                         szinesz += oneNode(subNode, "szinesz");
47                         studio += oneNode(subNode, "studio");
48                         film += oneNode(subNode, "film");
49                         szereples += oneNode(subNode, "szereples");
50                     }
51                 }
52             }
53         }
54         //Kimenet: String-ek
55         System.out.println(rendezo);
56         System.out.println(szinesz);
57         System.out.println(studio);
58         System.out.println(film);
59         System.out.println(szereples);
60     }
61
62     public static String oneNode(Node subNode, String tag) {
63         String out = "";
64         if (subNode.getNodeName().equals(tag)) { //tag String-ei megegyező nevű node-ot keres
65             if (subNode.getAttributes().getLength() > 0) //ha van attribute-ja akkor kiírja a legeleszt a String-be
66             {
67                 out += tag + " ID : " + subNode.getAttributes().item(0).getTextContent() + "\n";
68             }
69             NodeList subSubNodeList = subNode.getChildNodes();
70             for (int k = 0; k < subSubNodeList.getLength(); k++) {
71                 Node subSubNode = subSubNodeList.item(k); //root gyerek elemek egyessével
72                 if (subSubNode.getNodeType() == Node.ELEMENT_NODE) {
73                     switch (subSubNode.getNodeName()) { //az elem nevével függően máshogy rakja bele a String-be
74                         case "RID":
75                         case "SZID":
76                         case "SID":
77                         case "FID":
78                             out += subSubNode.getNodeName() + " : " + subSubNode.getAttributes().item(0).getTextContent() + "\n"; //az első attribute-omat rakja bele a String-be
79                             break;
80                         case "megjelenes":
81                         case "lakhely":
82                         case "szekhely":
83                             NodeList sssNode = subSubNode.getChildNodes();
84                             for (int l = 0; l < sssNode.getLength(); l++) { //az elem nevével függően máshogy rakja bele a String-be
85                                 if (sssNode.item(l).getNodeType() == Node.ELEMENT_NODE) {
86                                     out += subSubNode.getNodeName() + " : " + sssNode.item(l).getNodeName() + " : " + sssNode.item(l).getTextContent() + "\n";
87                                 }
88                             }
89                             break;
90                         default:
91                             out += subSubNode.getNodeName() + " : " + subSubNode.getTextContent() + "\n"; //normál esetben az elem nevet és a kontextusát rakja bele a String-be
92                         }
93                     }
94                 }
95             }
96             out += "\n";
97         }
98         return out;
99     }
100 }
101 }
```

A kimenete:

```
1 Rendezők
2
3 rendezo ID : r01
4 nev : David Zucker
5 eletkor : 74
6 lakhely-orszag : USA
7 lakhely-varos : Los Angeles
8
9 rendezo ID : r02
10 nev : Peter Segal
11 eletkor : 58
12 lakhely-orszag : USA
13 lakhely-varos : Chicago
14
15 rendezo ID : r03
16 nev : Norman Ivory Mayans
17 eletkor : 63
18 lakhely-orszag : USA
19 lakhely-varos : New York
20
21
22 Színeszek
23
24 színesz ID : sz01
25 nev : Leslie Nielsen
26 eletkor : 84
27 lakhely-orszag : USA
28 lakhely-varos : Fort Lauderdale
29
30 színesz ID : sz02
31 nev : Priscilla Presley
32 eletkor : 76
33 lakhely-orszag : USA
34 lakhely-varos : Los Angeles
35
36 színesz ID : sz03
37 nev : Marlon Mayans
38 eletkor : 49
39 lakhely-orszag : USA
40 lakhely-varos : Los Angeles
41
42
43 Stúdiók
44
45 studio ID : s01
46 nev : Paramount Pictures
47 alapitva : 1912
48 vezető : Adolf Zukor
49 színhely-orszag : USA
50 színhely-varos : Los Angeles
51
52 studio ID : s02
53 nev : Miramax
54 alapitva : 1979
55 vezető : Bob Weinstein
56 színhely-orszag : USA
57 színhely-varos : Santa Monica
58
59
60 Filmek
61
62 film ID : f01
63 cím : Caspaz piratoly
64 hossz : 78756177
65 hossz : 85
66 megjelenes-év : 1988
67 megjelenes-honap : 12
68 megjelenes-nap : 2
69 RID : r01
70 SID : s01
71
72 film ID : f02
73 cím : Caspaz piratoly 2 és 1/2
74 hossz : 152600000
75 hossz : 81
76 megjelenes-év : 1991
77 megjelenes-honap : 6
78 megjelenes-nap : 28
79 RID : r01
80 SID : s01
81
82 film ID : f03
83 cím : Caspaz piratoly 33 1/3 - Az utolsó merénylet
84 hossz : 51132596
85 hossz : 81
86 megjelenes-év : 1994
87 megjelenes-honap : 3
88 megjelenes-nap : 18
89 RID : r02
90 SID : s01
91
92 film ID : f04
93 cím : Horrorva átadva 2
94 hossz : 71300997
95 hossz : 83
96 megjelenes-év : 2001
97 megjelenes-honap : 7
98 megjelenes-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 : Kurtz Meeks
```



## 2b) Adatmódosítás

A kód:

```
1 package hu.domparse.nqixt9;
2
3 import java.io.File;
4 import java.io.IOException;
5 import java.io.UnsupportedEncodingException;
6 import java.util.InputMismatchException;
7 import java.util.Scanner;
8 import javax.xml.parsers.DocumentBuilder;
9 import javax.xml.parsers.DocumentBuilderFactory;
10 import javax.xml.parsers.ParserConfigurationException;
11 import javax.xml.transform.Transformer;
12 import javax.xml.transform.TransformerException;
13 import javax.xml.transform.TransformerFactory;
14 import javax.xml.transform.dom.DOMSource;
15 import javax.xml.transform.stream.StreamResult;
16 import org.w3c.dom.Document;
17 import org.w3c.dom.Node;
18 import org.w3c.dom.NodeList;
19 import org.xml.sax.SAXException;
20
21 public class DOMModifyNQIXT9 {
22
23     public static void main(String[] args){
24         // TODO: add your code here
25         DocumentBuilderFactory documentBuilderFactory = DocumentBuilderFactory.newInstance();
26
27         try {
28             DocumentBuilder documentBuilder = documentBuilderFactory.newDocumentBuilder();
29             Document document = documentBuilder.parse(new File("XMLNQIXT9.xml"));
30             document.getDocumentElement().normalize(); //normalizálás
31             Scanner scanner = new Scanner(System.in);
32             boolean exit=true;
33             while(exit){
34                 System.out.println("1 : Adj a meg egy film bevetelt\n2 : Adj a meg egy színész életkorát\n3 : Adj a meg egy studio új vezetőjének a nevét");
35                 switch(scanner.nextInt()){
36                     case 1 : fileIncome(document, scanner);break;
37                     case 2 : szinészAge(document, scanner);break;
38                     case 3 : studioBoss(document, scanner);break;
39                     default : exit=false;
40                 }
41             }
42             scanner.close();
43             writeToXml(document);
44         } catch (ParserConfigurationException | SAXException | IOException | TransformerException | InputMismatchException e) {
45             e.printStackTrace();
46         }
47     }
48
49     public static void fileIncome(Document document, Scanner scanner) {
50         NodeList nodeList = document.getElementsByTagName("film");
51         System.out.println("Adj a meg a film ID-ját :");
52         String input = scanner.next();
53         for (int i = 0; i < nodeList.getLength(); i++) {
54             Node node = nodeList.item(i);
55             if (node.getNodeType() == Node.ELEMENT_NODE) {
56                 if (node.getAttribute().getNamedItem("FID").getTextContent().equals(input)) {
57                     NodeList subNodeList = node.getChildNodes();
58                     for (int j = 0; j < subNodeList.getLength(); j++) {
59                         Node subNode = subNodeList.item(j);
60                         if (subNode.getNodeName().equals("bevetel")) {
61                             System.out.println("Adj a meg a film bevetelt :");
62                             subNode.setTextContent(scanner.next());
63                         }
64                     }
65                 }
66             }
67         }
68     }
69
70     public static void szinészAge(Document document, Scanner scanner) {
71         NodeList nodeList = document.getElementsByTagName("szinész");
72         System.out.println("Adj a meg a színész ID-ját :");
73         String input = scanner.next();
74         for (int i = 0; i < nodeList.getLength(); i++) {
75             Node node = nodeList.item(i);
76             if (node.getNodeType() == Node.ELEMENT_NODE) {
77                 if (node.getAttribute().getNamedItem("SzID").getTextContent().equals(input)) {
78                     NodeList subNodeList = node.getChildNodes();
79                     for (int j = 0; j < subNodeList.getLength(); j++) {
80                         Node subNode = subNodeList.item(j);
81                         if (subNode.getNodeName().equals("eletkor")) {
82                             System.out.println("Adj a meg a színész új életkorát :");
83                             subNode.setTextContent(scanner.next());
84                         }
85                     }
86                 }
87             }
88         }
89     }
90
91     public static void studioBoss(Document document, Scanner scanner) {
92         NodeList nodeList = document.getElementsByTagName("studio");
93         System.out.println("Adj a meg a studio ID-ját :");
94         String input = scanner.next();
95         for (int i = 0; i < nodeList.getLength(); i++) {
96             Node node = nodeList.item(i);
97             if (node.getNodeType() == Node.ELEMENT_NODE) {
98                 if (node.getAttribute().getNamedItem("SID").getTextContent().equals(input)) {
99                     NodeList subNodeList = node.getChildNodes();
100                     for (int j = 0; j < subNodeList.getLength(); j++) {
101                         Node subNode = subNodeList.item(j);
102                         if (subNode.getNodeName().equals("vezeto")) {
103                             System.out.println("Adj a meg a vezető új nevét :");
104                             String name = scanner.next();
105                             name += scanner.nextLine();
106                             subNode.setTextContent(name);
107                         }
108                     }
109                 }
110             }
111         }
112     }
113
114     public static void writeToXml(Document document) throws TransformerException, UnsupportedEncodingException {
115         TransformerFactory transformerFactory = TransformerFactory.newInstance();
116         Transformer transf = transformerFactory.newTransformer();
117         DOMSource source = new DOMSource(document);
118         StreamResult console = new StreamResult(System.out); // A módosítások kiírása a képernyőre
119         StreamResult file = new StreamResult(new File("XMLNQIXT9.out.xml")); // A módosítások kiírása új fájlba
120         transf.transform(source, console);
121         transf.transform(source, file);
122     }
123 }
```

# A kimenet:

```
1 : Adja meg egy film bevetelét
2 : Adja meg egy színész életkorát
3 : Adja meg egy stúdió új vezetőjének a nevét
4 :
5 Adja meg a film ID-jét :
6 FBI
7 Adja meg a film bevetelét :
8 8000000
9 : Adja meg egy film bevetelét
10 : Adja meg egy színész életkorát
11 : Adja meg egy stúdió új vezetőjének a nevét
12 :
13 Adja meg a színész ID-jét :
14 101
15 Adja meg a színész új életkorát :
16 30
17 : Adja meg egy film bevetelét
18 : Adja meg egy színész életkorát
19 : Adja meg egy stúdió új vezetőjének a nevét
20 :
21 Adja meg a stúdió ID-jét :
22 802
23 Adja meg a vezető új nevét :
24 Kiss Béla András
25 : Adja meg egy film bevetelét
26 : Adja meg egy színész életkorát
27 : Adja meg egy stúdió új vezetőjének a nevét
28 :
29 <?xml version="1.0" encoding="UTF-8" standalone="no"?><!--model_href="XMLSchemaQ12379.xsd" type="app:rootItem/ml" schemaLocation="http://www.w3.org/2001/XMLSchema?reflistars
30
31 <rendező RID="r1">
32 <nev>David Zucker</nev>
33 <életkor>74</életkor>
34 <labhely>
35 <ország>USA</ország>
36 <varos>Los Angeles</varos>
37 </labhely>
38
39 </rendező>
40 <rendező RID="r2">
41 <nev>Peter Segal</nev>
42 <életkor>58</életkor>
43 <labhely>
44 <ország>USA</ország>
45 <varos>Chicago</varos>
46 </labhely>
47
48 </rendező>
49 <rendező RID="r3">
50 <nev>Kenny Burg Myers</nev>
51 <életkor>63</életkor>
52 <labhely>
53 <ország>USA</ország>
54 <varos>New York</varos>
55 </labhely>
56 </rendező>
57
58 <színesztárs SID="s1">
59 <nev>Leslie Nielsen</nev>
60 <életkor>89</életkor>
61 <labhely>
62 <ország>USA</ország>
63 <varos>Fort Lauderdale</varos>
64 </labhely>
65
66 </színesztárs>
67 <színesztárs SID="s2">
68 <nev>Patricia Presley</nev>
69 <életkor>76</életkor>
70 <labhely>
71 <ország>USA</ország>
72 <varos>Los Angeles</varos>
73 </labhely>
74
75 </színesztárs>
76 <színesztárs SID="s3">
77 <nev>Marlon Wayans</nev>
78 <életkor>49</életkor>
79 <labhely>
80 <ország>USA</ország>
81 <varos>Los Angeles</varos>
82 </labhely>
83 </színesztárs>
84
85 <stúdió SID="s4">
86 <nev>Paramount Pictures</nev>
87 <elintézés>392</elintézés>
88 <vezető>Dobó Zoltán</vezető>
89 <labhely>
90 <ország>USA</ország>
91 <varos>Los Angeles</varos>
92 </labhely>
93
94 </stúdió>
95 <stúdió SID="s5">
96 <nev>Miramax</nev>
97 <elintézés>379</elintézés>
98 <vezető>Kiss Béla András</vezető>
99 <labhely>
100 <ország>USA</ország>
101 <varos>Santa Monica</varos>
102 </labhely>
103 </stúdió>
104
105 <film FID="f1">
106 < cíle>Csupasz pisztoly</cíle>
107 <nev>Csupasz pisztoly</nev>
108 <chassi>45</chassi>
109 <magjelenés>
110 <ev>1998</ev>
111 <chassp>32</chassp>
112 <magj>2</magj>
113 </magjelenés>
114 <OID refID="r1"/>
115 <OID refID="s1"/>
116 </film>
117
118 <film FID="f2">
119 < cíle>Csupasz pisztoly 2 és 1/2</cíle>
120 <nev>Csupasz pisztoly 2 és 1/2</nev>
121 <chassi>45</chassi>
122 <magjelenés>
123 <ev>1994</ev>
124 <chassp>36</chassp>
125 <magj>3</magj>
126 </magjelenés>
127 <OID refID="r2"/>
128 <OID refID="s2"/>
129 </film>
130
131 <film FID="f3">
132 < cíle>Hárverre akadva 2</cíle>
133 <nev>Hárverre akadva 2</nev>
134 <chassi>43</chassi>
135 <magjelenés>
136 <ev>2004</ev>
137 <chassp>36</chassp>
138 <magj>4</magj>
139 </magjelenés>
140 <OID refID="r3"/>
141 <OID refID="s3"/>
142 </film>
143
144 </film>
145
146 <szereplős SID="s1">
147 <OID refID="r1"/>
148 <OID refID="s1"/>
149 <szereplős>Frank Drabin hadnagy</szereplős>
150 </szereplős>
151
152 <szereplős SID="s2">
153 <OID refID="r1"/>
154 <OID refID="s2"/>
155 <szereplős>Jane Spencer</szereplős>
156 </szereplős>
157
158 <szereplős SID="s3">
159 <OID refID="r2"/>
160 <OID refID="s3"/>
161 <szereplős>Frank Drabin hadnagy</szereplős>
162 </szereplős>
163
164 <szereplős SID="s4">
165 <OID refID="r2"/>
166 <OID refID="s4"/>
167 <szereplős>Frank Drabin hadnagy</szereplős>
168 </szereplős>
169
170 <szereplős SID="s5">
171 <OID refID="r3"/>
172 <OID refID="s5"/>
173 <szereplős>Frank Drabin hadnagy</szereplős>
174 </szereplős>
175
176 <szereplős SID="s6">
177 <OID refID="r3"/>
178 <OID refID="s6"/>
179 <szereplős>Frank Drabin hadnagy</szereplős>
180 </szereplős>
181
182 <szereplős SID="s7">
183 <OID refID="r3"/>
184 <OID refID="s7"/>
185 <szereplős>Frank Drabin hadnagy</szereplős>
186 </szereplős>
187
188 </szereplős>
189 </film>
```

## 3c) Adatlekérdezés

A kód:

```
1 package hu.dongaris.q1c19;
2
3 import java.io.File;
4 import java.io.IOException;
5 import java.util.Calendar;
6 import java.util.Scanner;
7
8 import javax.xml.parsers.DocumentBuilder;
9 import javax.xml.parsers.DocumentBuilderFactory;
10 import javax.xml.parsers.ParserConfigurationException;
11 import org.w3c.dom.*;
12 import org.xml.sax.SAXException;
13
14 public class DOMQueryQ1C19 {
15
16     public static void main(String[] args){
17         //DOM tree
18         DocumentBuilderFactory documentBuilderFactory = DocumentBuilderFactory.newInstance();
19
20         try {
21             DocumentBuilder documentBuilder = documentBuilderFactory.newDocumentBuilder();
22             Document document = documentBuilder.parse(new File("xml/q1c19.xml"));
23             document.getDocumentElement().normalize();
24             Scanner scanner = new Scanner(System.in);
25             boolean exitFlag;
26             while(exit) {
27                 System.out.println("1 : Filmeszék másik több bevetelt hoztak mint... \n2 : Színházszék másik fiatalabbak... \n3 : Stúdiószék másik fiatalabbak mint...");
28                 switch(scanner.nextInt()) {
29                     case 1 : filmeszék(document, scanner); break;
30                     case 2 : színházszék(document, scanner); break;
31                     case 3 : stúdiószék(document, scanner); break;
32                     default : exitFlag = true;
33                 }
34                 scanner.close();
35             } catch (ParserConfigurationException | SAXException | IOException e) {
36                 e.printStackTrace();
37             }
38         }
39
40         public static void filmeszék(Document document, Scanner scanner) {
41             NodeList nodeList = document.getElementsByTagName("film");
42             System.out.println("Adja meg mennyit néz:");
43             Long input = scanner.nextLong();
44             System.out.println("\n");
45             for (int i = 0; i < nodeList.getLength(); i++) {
46                 Node node = nodeList.item(i);
47                 if (node.getNodeType() == Node.ELEMENT_NODE) {
48                     NodeList subNodeList = node.getChildNodes();
49                     for (int j = 0; j < subNodeList.getLength(); j++) {
50                         Node subNode = subNodeList.item(j);
51                         if (subNode.getNodeType() == Node.ELEMENT_NODE) {
52                             if (subNode.getTagName().equals("bevetel")) {
53                                 if (Long.parseLong(subNode.getTextContent()) > input)
54                                     listSub(node, "film");
55                             }
56                         }
57                     }
58                 }
59             }
60
61             public static void színházszék(Document document, Scanner scanner) {
62                 NodeList nodeList = document.getElementsByTagName("színház");
63                 System.out.println("Adja meg mennyit néz:");
64                 int input = scanner.nextInt();
65                 System.out.println("\n");
66                 for (int i = 0; i < nodeList.getLength(); i++) {
67                     Node node = nodeList.item(i);
68                     if (node.getNodeType() == Node.ELEMENT_NODE) {
69                         NodeList subNodeList = node.getChildNodes();
70                         for (int j = 0; j < subNodeList.getLength(); j++) {
71                             Node subNode = subNodeList.item(j);
72                             if (subNode.getNodeType() == Node.ELEMENT_NODE) {
73                                 if (subNode.getTagName().equals("színház")) {
74                                     if (Integer.parseInt(subNode.getTextContent()) < input)
75                                         listSub(node, "színház");
76                                 }
77                             }
78                         }
79                     }
80                 }
81
82                 public static void stúdiószék(Document document, Scanner scanner) {
83                     NodeList nodeList = document.getElementsByTagName("stúdió");
84                     System.out.println("Adja meg mennyit néz:");
85                     Long input = scanner.nextLong();
86                     System.out.println("\n");
87                     int year = Calendar.getInstance().get(Calendar.YEAR);
88                     for (int i = 0; i < nodeList.getLength(); i++) {
89                         Node node = nodeList.item(i);
90                         if (node.getNodeType() == Node.ELEMENT_NODE) {
91                             NodeList subNodeList = node.getChildNodes();
92                             for (int j = 0; j < subNodeList.getLength(); j++) {
93                                 Node subNode = subNodeList.item(j);
94                                 if (subNode.getNodeType() == Node.ELEMENT_NODE) {
95                                     if (subNode.getTagName().equals("stúdió")) {
96                                         if (year - Integer.parseInt(subNode.getTextContent()) < input)
97                                             listSub(node, "stúdió");
98                                     }
99                                 }
100                             }
101                         }
102                     }
103                 }
104
105                 private static void listSub(Node node, String tag) {
106                     String out = "";
107                     out += tag + " ID : " + node.getAttribute("id").getTextContent() + "\n";
108                     Element element = (Element) node;
109                     switch (tag) {
110                         case "film":
111                             out += element.getAttribute("id") + " : " + element.getAttribute("cím").getTextContent() + "\n";
112                             out += element.getAttribute("cím").item(0).getTextContent() + "\n";
113                             out += element.getAttribute("bevetel").item(0).getTextContent() + "\n";
114                             break;
115                         case "színház":
116                             out += element.getAttribute("id") + " : " + element.getAttribute("cím").getTextContent() + "\n";
117                             out += element.getAttribute("cím").item(0).getTextContent() + "\n";
118                             out += element.getAttribute("elektor").item(0).getTextContent() + "\n";
119                             break;
120                         case "stúdió":
121                             out += element.getAttribute("id") + " : " + element.getAttribute("cím").getTextContent() + "\n";
122                             out += element.getAttribute("cím").item(0).getTextContent() + "\n";
123                             out += element.getAttribute("alapítás").item(0).getTextContent() + "\n";
124                             break;
125                     }
126                     System.out.println(out);
127                 }
128             }
129         }
130     }
131 }
```

## 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 bevétel : 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 bevétel : 78756177
24
25 Film ID : f02
26 cím : Csupasz pisztoly 2 és 1/2
27 bevétel : 192000000
28
29 Film ID : f03
30 cím : Csupasz pisztoly 33 1/3 - Az utolsó merénylet
31 bevétel : 51132598
32
33 Film ID : f04
34 cím : Horrorra akadva 2
35 bevétel : 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 alapítás : 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
```