

# JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Kézilabda adatbázis kezelése XML-ben

Készítette: **Torma Antal**

Neptunkód: **YK11Q1**

## Tartalomjegyzék

A feladat leírása.....	3
1. Feladat .....	3
1.1 Az adatbázis ER modellje .....	3
1.2 Az ER modell konvertálása XDM modellre .....	3
1.3 Az XDM modell alapján validált XML dokumentum készítése .....	4
1.4 Az XML dokumentum alapján XMLSchema készítése .....	9
2. Feladat .....	12
2.1 Adatolvasás .....	12
2.2 Adatlekérdezés .....	16
2.3 Adatmódosítás .....	18

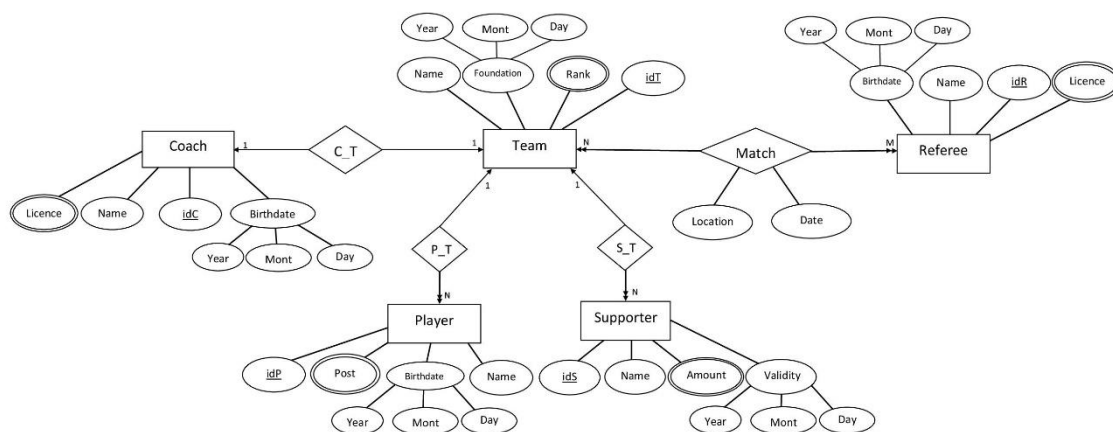
**A feladat leírása:** A tárgy féléves feladatát két részre bonthatjuk. Az első rész célja, hogy a hallgató létrehozzon az oktatóval előre megbeszélt témában egy adatbázis ER modellt, amit átkonvertál XDM modellre. Majd ezt követően elkészíti az XDM modell alapján validált XML dokumentum és a hozzá tartozó XML Schema-t.

A feladat második része az XML dokumentum olvasása, lekérdezések készítése és az adatok módosítása. Java programozási nyelven, azon belül a DOM parser használata volt kötelező, ami az XML fájlt a memóriában fa szerkezetben tárolja

## 1. Feladat

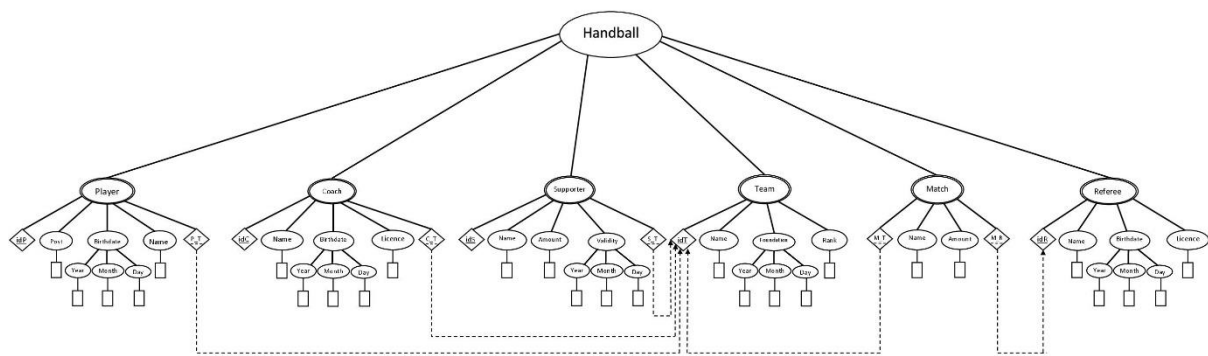
### 1.1 Az adatbázis ER modellje

Az ER modell 5 egyedet tartalmaz melyek téglalappal vannak jelölve. Az egyedek kapcsolatban állnak más egyedekkel, amiket a rombusz alakzatok reprezentálnak. A kapcsolatok lehetnek 1:1, 1:N és N:M számosságúak, az egyszeri kapcsolat teli nyíllal míg a többszöri dupla teli nyíllal köti össze az egyedeket. Továbbá az egyedek többfajta tulajdonsággal rendelkezhetnek, lehet egyszerű(ellipszis), többértékű (dupla ellipszis), összetett (saját tulajdonsága van), kulcs (ellipszis aláhúzással). A modell megtervezéséhez a drawio-t használtam.



### 1.2 Az ER modell konvertálása XDM modellre

A konvertálás után az egyedeket elemeknek nevezzük és ellipszissel jelöljük. A XDM modell tetején a gyöker elem áll, amikhez csatlakoznak a gyerek elemek. A többszörösen előfordulható elemeknél dupla ellipszist használunk, az elemek értékeit pedig oldalára állított téglalappal jelezzük. Azon kapcsolatok, amelyek tulajdonságokkal rendelkeztek a továbbiakban különálló elemként jelöljük. A kulcsokat és az idegen kulcsokat rombuszként előbbit aláhúzással, utóbbit szaggatott aláhúzással jelenítünk meg. Az idegen kulcsok szaggatott vonallal a végén teli nyíllal mutatnak egy másik elem kulcs tulajdonságára. A modell megtervezéséhez a drawio-t használtam.



### 1.3 Az XDM modell alapján validált XML dokumentum készítése

Az XML dokumentum fa szerkezetű, minden elemhez tartozik egy nyitó és záró tag, a gyökerelem taggal kezdődik és záródik és ezen belül helyezkednek el a gyerekelemei. A kulcsok, idegenkulcsok az elemek attribútumaiként jelennek meg, az értékeket pedig tartalommal töltöttem fel. Minden többszörösen előfordulható elemből 5 példányt készítettem. Az XML dokumentumot az <oXygen/> XML Editor 21.0 program használatával hoztam létre. A felhasznált adatok egy része fiktív, de tartalmaz valós adatokat is.

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <handball xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3      xsi:noNamespaceSchemaLocation="XMLSchemaYk1lq1.xsd">
4      <!-- Öt különböző team példány egyedi ID-val és adatokkal-->
5      <team idT="t1">
6          <name>DVTK</name>
7          <foundation>
8              <year>1910</year>
9              <month>01</month>
10             <day>01</day>
11         </foundation>
12         <rank>NBII</rank>
13     </team>
14     <team idT="t2">
15         <name>FLER</name>
16         <foundation>
17             <year>1986</year>
18             <month>06</month>
19             <day>15</day>
20         </foundation>
21         <rank>NB I/B</rank>
22     </team>
23     <team idT="t3">
24         <name>HTE</name>
25         <foundation>
26             <year>1919</year>
27             <month>06</month>
28             <day>01</day>
29         </foundation>
30         <rank>NBII</rank>
31     </team>

```

```

32 <team idT="t4">
33     <name>Veszprém</name>
34 <foundation>
35     <year>1977</year>
36     <month>03</month>
37     <day>19</day>
38 </foundation>
39     <rank>NBI</rank>
40 </team>
41 <team idT="t5">
42     <name>DEAC</name>
43 <foundation>
44     <year>1919</year>
45     <month>10</month>
46     <day>05</day>
47 </foundation>
48     <rank>NBI/B</rank>
49 </team>
50 <!-- Öt különböző supporter példány egyedi ID-val, adatokkal és idegenkulccsal-->
51 <supporter idS="s1" s_t="t1">
52     <name>Apollo Tyres</name>
53     <amount>100000000</amount>
54 <validity>
55     <year>2025</year>
56     <month>06</month>
57     <day>31</day>
58 </validity>
59 </supporter>
60 <supporter idS="s2" s_t="t4">
61     <name>Telekom</name>
62     <amount>100000000</amount>
63 <validity>
64     <year>2030</year>
65     <month>12</month>
66     <day>31</day>
67 </validity>
68 </supporter>
69 <supporter idS="s3" s_t="t3">
70     <name>Adidas</name>
71     <amount>100000000</amount>
72 <validity>
73     <year>2024</year>
74     <month>01</month>
75     <day>01</day>
76 </validity>
77 </supporter>
78 <supporter idS="s4" s_t="t5">
79     <name>Hall</name>
80     <amount>5000000</amount>
81 <validity>
82     <year>2029</year>
83     <month>12</month>
84     <day>01</day>
85 </validity>
86 </supporter>

```

```

87 <supporter idS="s5" s_t="t2">
88     <name>Erima</name>
89     <amount>1000000</amount>
90     <validity>
91         <year>2030</year>
92         <month>10</month>
93         <day>15</day>
94     </validity>
95 </supporter>
96 <!-- Öt player példány egyedi ID-val, adatokkal és idegenkulccsal -->
97 <player idP="p1" p_t="t1">
98     <name>Torma Antal</name>
99     <birthdate>
100         <year>1994</year>
101         <month>09</month>
102         <day>02</day>
103     </birthdate>
104     <post>Kapus</post>
105 </player>
106 <player idP="p2" p_t="t3">
107     <name>Kiss András</name>
108     <birthdate>
109         <year>1970</year>
110         <month>03</month>
111         <day>11</day>
112     </birthdate>
113     <post>Beállító</post>
114 </player>
115 <player idP="p3" p_t="t2">
116     <name>Nagy Róbert</name>
117     <birthdate>
118         <year>1999</year>
119         <month>04</month>
120         <day>30 </day>
121     </birthdate>
122     <post>Irányító</post>
123 </player>
124 <player idP="p4" p_t="t5">
125     <name>Győrfi Ádám</name>
126     <birthdate>
127         <year>2000</year>
128         <month>02</month>
129         <day>23</day>
130     </birthdate>
131     <post>Szélső</post>
132 </player>
133 <player idP="p5" p_t="t4">
134     <name>Dávid Máté</name>
135     <birthdate>
136         <year>1992</year>
137         <month>07</month>
138         <day>30</day>
139     </birthdate>
140     <post>Átlövő</post>
141 </player>

```

```
142 <!-- Öt coach példány egyedi ID-val, adatokkal és idegenkulccsal -->
143 <coach idC="c1" c_t="t5">
144     <name>Gipsz Jakab</name>
145     <birthdate>
146         <year>1980</year>
147         <month>05</month>
148         <day>26</day>
149     </birthdate>
150     <licence>C</licence>
151 </coach>
152 <coach idC="c2" c_t="t1">
153     <name>Molnár András</name>
154     <birthdate>
155         <year>1970</year>
156         <month>07</month>
157         <day>11</day>
158     </birthdate>
159     <licence>A</licence>
160 </coach>
161 <coach idC="c3" c_t="t2">
162     <name>Tamás Tamás</name>
163     <birthdate>
164         <year>1992</year>
165         <month>12</month>
166         <day>01</day>
167     </birthdate>
168     <licence>B</licence>
169 </coach>
170 <coach idC="c4" c_t="t3">
171     <name>Bakos Károly</name>
172     <birthdate>
173         <year>1989</year>
174         <month>08</month>
175         <day>21</day>
176     </birthdate>
177     <licence>B</licence>
178 </coach>
179 <coach idC="c5" c_t="t4">
180     <name>Momir Ilic</name>
181     <birthdate>
182         <year>1986</year>
183         <month>01</month>
184         <day>30</day>
185     </birthdate>
186     <licence>B</licence>
187 </coach>
```

```

188 <!-- Öt match példány melyek kettő idegenkulcsot tartalmazznak -->
189 <match m_t="t2" m_r="r1">
190     <location>Budapest</location>
191     <date>2022-10-31</date>
192 </match>
193 <match m_t="t1" m_r="r3">
194     <location>Miskolc</location>
195     <date>2022-11-19</date>
196 </match>
197 <match m_t="t3" m_r="r2">
198     <location>Hajdúböszörmény</location>
199     <date>2022-12-02</date>
200 </match>
201 <match m_t="t5" m_r="r4">
202     <location>Veszprém</location>
203     <date>2023-03-11</date>
204 </match>
205 <match m_t="t4" m_r="r5">
206     <location>Debrecen</location>
207     <date>2023-05-20</date>
208 </match>
209 <!-- Öt referee példány ID-val és adatokkal -->
210 <referee idR="r1">
211     <name>Katona János</name>
212     <birthdate>
213         <year>1975</year>
214         <month>01</month>
215         <day>05</day>
216     </birthdate>
217     <licence>C</licence>
218 </referee>
219 <referee idR="r2">
220     <name>Tóth István</name>
221     <birthdate>
222         <year>1972</year>
223         <month>09</month>
224         <day>22</day>
225     </birthdate>
226     <licence>A</licence>
227 </referee>
228 <referee idR="r3">
229     <name>Nagy Péter</name>
230     <birthdate>
231         <year>1993</year>
232         <month>11</month>
233         <day>11</day>
234     </birthdate>
235     <licence>A</licence>
236 </referee>

```



```

237 <referee idR="r4">
238   <name>Bíró Ádám</name>
239   <birthdate>
240     <year>1990</year>
241     <month>04</month>
242     <day>16</day>
243   </birthdate>
244   <licence>C</licence>
245 </referee>
246 <referee idR="r5">
247   <name>Kiss Olivér</name>
248   <birthdate>
249     <year>1988</year>
250     <month>11</month>
251     <day>27</day>
252   </birthdate>
253   <licence>C</licence>
254 </referee>
255 </handball>

```

#### 1.4 Az XML dokumentum alapján XMLSchema készítése

Komplex típusként hoztam létre azon gyerekelemeket, amelyeknek további gyerekelei lehetnek. A többértékű elemek pedig egyszerű típusként reprezentáltam. Minden kulcs és idegen kulcs értéke egyedi és kötelező, továbbá létrehoztam a kapcsolatokat a kulcsok között. A Schema-t az <oXygen/> XML Editor 21.0 program segítségével készítettem el.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
3
4   <!-- Szerkezet felépítése -->
5   <xs:element name="handball">
6     <xs:complexType>
7       <xs:sequence>
8         <xs:element name="team" type="teamTipus" maxOccurs="unbounded"/>
9         <xs:element name="supporter" type="supporterTipus" maxOccurs="unbounded"/>
10        <xs:element name="player" type="playerTipus" maxOccurs="unbounded"/>
11        <xs:element name="coach" type="coachTipus" maxOccurs="unbounded"/>
12        <xs:element name="match" type="matchTipus" maxOccurs="unbounded"/>
13        <xs:element name="referee" type="refereeTipus" maxOccurs="unbounded"/>
14      </xs:sequence>
15    </xs:complexType>
16
17    <!-- PK kulcsok definiálása -->
18    <xs:key name="player_kulcs">
19      <xs:selector xpath="player"/>
20      <xs:field xpath="@idP"/>
21    </xs:key>
22    <xs:key name="coach_kulcs">
23      <xs:selector xpath="coach"/>
24      <xs:field xpath="@idC"/>
25    </xs:key>
26    <xs:key name="team_kulcs">
27      <xs:selector xpath="team"/>
28      <xs:field xpath="@idT"/>
29    </xs:key>
30    <xs:key name="supporter_kulcs">
31      <xs:selector xpath="supporter"/>
32      <xs:field xpath="@idS"/>
33    </xs:key>
34    <xs:key name="referee_kulcs">
35      <xs:selector xpath="referee"/>
36      <xs:field xpath="@idR"/>
37    </xs:key>
38

```

```

39      <!-- FK kulcsok definiálása -->
40      <xs:keyref refer="team_kulcs" name="team_idegen_kulcs">
41          <xs:selector xpath="match"/>
42          <xs:field xpath="@m_t"/>
43      </xs:keyref>
44      <xs:keyref refer="referee_kulcs" name="referee_idegen_kulcs">
45          <xs:selector xpath="match"/>
46          <xs:field xpath="@m_r"/>
47      </xs:keyref>
48      <xs:keyref refer="team_kulcs" name="team_coach_idegen_kulcs">
49          <xs:selector xpath="coach"/>
50          <xs:field xpath="@c_t"/>
51      </xs:keyref>
52      <xs:keyref refer="team_kulcs" name="team_player_idegen_kulcs">
53          <xs:selector xpath="player"/>
54          <xs:field xpath="@p_t"/>
55      </xs:keyref>
56      <xs:keyref refer="team_kulcs" name="team_supporter_idegen_kulcs">
57          <xs:selector xpath="supporter"/>
58          <xs:field xpath="@s_t"/>
59      </xs:keyref>
60
61      <!-- 1:1 kapcsolat létrehozása -->
62      <xs:unique name="unique_coach">
63          <xs:selector xpath="coach"></xs:selector>
64          <xs:field xpath="@c_t"></xs:field>
65      </xs:unique>
66
67  </xs:element>
68
69  <!-- Egyszerű típusok definiálása -->
70  <xs:simpleType name="rankTipus">
71      <xs:restriction base="xs:string">
72          <xs:enumeration value="NBI"/>
73          <xs:enumeration value="NBI/B"/>
74          <xs:enumeration value="NBII"/>
75          <xs:enumeration value="Megye"/>
76      </xs:restriction>
77  </xs:simpleType>
78
79  <xs:simpleType name="postTipus">
80      <xs:restriction base="xs:string">
81          <xs:enumeration value="Kapus"/>
82          <xs:enumeration value="Szélső"/>
83          <xs:enumeration value="Beállító"/>
84          <xs:enumeration value="Írányító"/>
85          <xs:enumeration value="Átlövő"/>
86      </xs:restriction>
87  </xs:simpleType>
88
89  <xs:simpleType name="licenceTipus">
90      <xs:restriction base="xs:string">
91          <xs:enumeration value="A"/>
92          <xs:enumeration value="B"/>
93          <xs:enumeration value="C"/>
94          <xs:enumeration value="D"/>
95      </xs:restriction>
96  </xs:simpleType>
97
98  <xs:simpleType name="amountTipus">
99      <xs:restriction base="xs:integer">
100          <xs:enumeration value="1000000"/>
101          <xs:enumeration value="5000000"/>
102          <xs:enumeration value="10000000"/>
103          <xs:enumeration value="50000000"/>
104          <xs:enumeration value="100000000"/>
105      </xs:restriction>
106  </xs:simpleType>
107
108

```

```

109 <!-- Komplex típusok definiálása -->
110 <xs:complexType name="dateTípus">
111   <xs:sequence>
112     <xs:element name="year">
113       <xs:simpleType>
114         <xs:restriction base="xs:integer">
115           <xs:totalDigits value="4"/>
116         </xs:restriction>
117       </xs:simpleType>
118     </xs:element>
119     <xs:element name="month">
120       <xs:simpleType>
121         <xs:restriction base="xs:integer">
122           <xs:totalDigits value="2"/>
123         </xs:restriction>
124       </xs:simpleType>
125     </xs:element>
126     <xs:element name="day">
127       <xs:simpleType>
128         <xs:restriction base="xs:integer">
129           <xs:totalDigits value="2"/>
130         </xs:restriction>
131       </xs:simpleType>
132     </xs:element>
133   </xs:sequence>
134 </xs:complexType>
135
136 <xs:complexType name="teamTípus">
137   <xs:sequence>
138     <xs:element name="name" type="xs:string"/>
139     <xs:element name="foundation" type="dateTípus"/>
140     <xs:element name="rank" type="rankTípus"/>
141   </xs:sequence>
142   <xs:attribute name="idT" type="xs:string" use="required"/>
143 </xs:complexType>
144
145 <xs:complexType name="supporterTípus">
146   <xs:sequence>
147     <xs:element name="name" type="xs:string"/>
148     <xs:element name="amount" type="amountTípus"/>
149     <xs:element name="validity" type="dateTípus"/>
150   </xs:sequence>
151   <xs:attribute name="idS" type="xs:string" use="required"/>
152   <xs:attribute name="s_t" type="xs:string" use="required"/>
153 </xs:complexType>
154
155 <xs:complexType name="playerTípus">
156   <xs:sequence>
157     <xs:element name="name" type="xs:string"/>
158     <xs:element name="birthdate" type="dateTípus"/>
159     <xs:element name="post" type="postTípus"/>
160   </xs:sequence>
161   <xs:attribute name="idP" type="xs:string" use="required"/>
162   <xs:attribute name="p_t" type="xs:string" use="required"/>
163 </xs:complexType>
164
165 <xs:complexType name="coachTípus">
166   <xs:sequence>
167     <xs:element name="name" type="xs:string"/>
168     <xs:element name="birthdate" type="dateTípus"/>
169     <xs:element name="licence" type="licenceTípus"/>
170   </xs:sequence>
171   <xs:attribute name="idC" type="xs:string" use="required"/>
172   <xs:attribute name="c_t" type="xs:string" use="required"/>
173 </xs:complexType>
174
175 <xs:complexType name="matchTípus">
176   <xs:sequence>
177     <xs:element name="location" type="xs:string"/>
178     <xs:element name="date" type="xs:date"/>
179   </xs:sequence>
180   <xs:attribute name="m_t" type="xs:string" use="required"/>
181   <xs:attribute name="m_r" type="xs:string" use="required"/>
182 </xs:complexType>
183

```

## 2. Feladat

### 2.1 Adatolvasás

xml.parsers.DocumentBuilder és xml.parsers.DocumentBuilderFactory osztályok használatával az XML dokumentumból létrehoztam egy DOM fájlt amely fa szerkezetben reprezentálja az XML adatokat. A gyöker elem és az elem csomópontok lekérdezésével hozzáférünk az elemek tartalmához, amelyet feldolgozás után strukturált formában tudunk kiírni konzolra.

```
1 package hu.domp.parse.yk11q1;
2
3 import org.w3c.dom.Document;
4 import org.w3c.dom.Element;
5 import org.w3c.dom.Node;
6 import org.w3c.dom.NodeList;
7 import org.xml.sax.SAXException;
8
9 import javax.xml.parsers.DocumentBuilder;
10 import javax.xml.parsers.DocumentBuilderFactory;
11 import javax.xml.parsers.ParserConfigurationException;
12 import java.io.File;
13 import java.io.FileWriter;
14 import java.io.IOException;
15
16 public class DomReadYk11q1 {
17     public static void main(String[] args) throws SAXException, IOException, ParserConfigurationException {
18         //XML fájl betöltése
19         File xmlFile = new File("XMLYk11q1.xml");
20
21         //Egy txt fájl létrehozása amibe kiíratjuk az XML fájl adatait
22         FileWriter fileWriter = new FileWriter("XMLYk11q1.txt");
23
24         //DocumentBuilderFactory példányosítása a newInstance() metódussal, ebből megkapjuk a DocumentBuildert
25         DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
26         DocumentBuilder dBuilder = factory.newDocumentBuilder();
27
28         //A parse() metódussal megkapjuk a DOM dokumentumot az XML-ből
29         Document doc = dBuilder.parse(xmlFile);
30
31         //Dokumentum normalizálása
32         doc.getDocumentElement().normalize();
33
34         //Gyöker elem kiírása
35         System.out.println("Root element: " + doc.getDocumentElement().getNodeName() + "\n");
36         System.out.println("-----");
37
38         //Team elem nodelistjének lekérdezése
39         NodeList nList = doc.getElementsByTagName("team");
40
41         //Végig megyünk a listán
42         for (int i = 0; i < nList.getLength(); i++) {
43             Node tNode = nList.item(i);
44             System.out.println("Current element: " + tNode.getNodeName());
45             if (tNode.getNodeType() == Node.ELEMENT_NODE) {
46
47                 //Elementé alakítjuk az aktuális elemet
48                 Element elem = (Element) tNode;
49
50                 //Lekérjük az aktuális elem attribútumának és gyerekelemeinek tartalmát
51                 String idT = elem.getAttribute("idT");
52                 String name = elem.getElementsByTagName("name").item(0).getTextContent();
53                 NodeList foundationList = elem.getElementsByTagName("foundation");
54                 String foundation = "";
```

```

55     for (int j = 0; j < foundationList.getLength(); j++) {
56         Node dNode = foundationList.item(j);
57         if (dNode.getNodeType() == Node.ELEMENT_NODE) {
58             Element dElem = (Element) dNode;
59             foundation = dElem.getElementsByTagName("year").item( index: 0).getTextContent();
60             foundation += "-" + dElem.getElementsByTagName("month").item( index: 0).getTextContent();
61             foundation += "-" + dElem.getElementsByTagName("day").item( index: 0).getTextContent();
62         }
63     }
64     String rank = elem.getElementsByTagName("rank").item( index: 0).getTextContent();
65
66     //Adatok kiírása konzolra és a txt fájlba
67     System.out.println("Team ID: " + idT);
68     System.out.println("Name: " + name);
69     System.out.println("Foundation: " + foundation);
70     System.out.println("Rank: " + rank + "\n");
71     fileWriter.write( str: "Team ID: " + idT + "\nName: " + name + "\nFoundation: " + foundation + "\nRank: " + rank + "\n\n");
72 }
73
74 }
75
76 System.out.println("-----\n");
77 fileWriter.write( str: "\n-----\n");
78 //Supporter elem nodelistjének lekérdezése
79 nList = doc.getElementsByTagName("supporter");
80
81 //Végig megyünk a listán
82 for (int i = 0; i < nList.getLength(); i++) {
83     Node sNode = nList.item(i);
84     System.out.println("Current element: " + sNode.getNodeName());
85     if (sNode.getNodeType() == Node.ELEMENT_NODE) {
86
87         // Elementé alakítjuk az aktuális elemet
88         Element elem = (Element) sNode;
89
90         //Lekérjük az aktuális elem atributeumának és gyerekelemeinek tartalmát
91         String idT = elem.getAttribute( name: "idS");
92         String s_t = elem.getAttribute( name: "s_t");
93         String name = elem.getElementsByTagName("name").item( index: 0).getTextContent();
94         String amount = elem.getElementsByTagName("amount").item( index: 0).getTextContent();
95         NodeList nValidity = elem.getElementsByTagName("validity");
96         String validity = "";
97         for (int j = 0; j < nValidity.getLength(); j++) {
98             Node dNode = nValidity.item(j);
99             if (dNode.getNodeType() == Node.ELEMENT_NODE) {
100                 Element dElem = (Element) dNode;
101                 validity = dElem.getElementsByTagName("year").item( index: 0).getTextContent();
102                 validity += "-" + dElem.getElementsByTagName("month").item( index: 0).getTextContent();
103                 validity += "-" + dElem.getElementsByTagName("day").item( index: 0).getTextContent();
104             }
105         }
106
107         //Adatok kiírása konzolra és a txt fájlba
108         System.out.println("Supporter ID: " + idT);
109         System.out.println("S_T FK: " + s_t);
110         System.out.println("Name: " + name);
111         System.out.println("Amount: " + amount);
112         System.out.println("Validity: " + validity + "\n");
113         fileWriter.write( str: "Supporter ID: " + idT + "\nS_T FK: " + s_t + "\nName: " + name + "\nAmount: " + amount + "\nValidity: " + validity + "\n\n");
114     }
115 }
116
117 System.out.println("-----\n");
118 fileWriter.write( str: "\n-----\n");
119
120 //Supporter elem nodelistjének lekérdezése
121 nList = doc.getElementsByTagName("player");
122
123 //Végig megyünk a listán
124 for (int i = 0; i < nList.getLength(); i++) {
125     Node pNode = nList.item(i);
126     System.out.println("Current element: " + pNode.getNodeName());
127     if (pNode.getNodeType() == Node.ELEMENT_NODE) {
128
129         // Elementé alakítjuk az aktuális elemet
130         Element elem = (Element) pNode;
131
132         //Lekérjük az aktuális elem atributeumának és gyerekelemeinek tartalmát
133         String idP = elem.getAttribute( name: "idP");
134         String p_t = elem.getAttribute( name: "p_t");
135         String name = elem.getElementsByTagName("name").item( index: 0).getTextContent();
136         NodeList nBirthdate = elem.getElementsByTagName("birthdate");
137         String birthdate = "";

```

```

    for (int j = 0; j < nBirthdate.getLength(); j++) {
        Node dNode = nBirthdate.item(j);
        if (dNode.getNodeType() == Node.ELEMENT_NODE) {
            Element eDate = (Element) dNode;
            birthdate = eDate.getElementsByTagName("year").item( index: 0).getTextContent();
            birthdate += "-" + eDate.getElementsByTagName("month").item( index: 0).getTextContent();
            birthdate += "-" + eDate.getElementsByTagName("day").item( index: 0).getTextContent();
        }
    }

    String post = eElem.getElementsByTagName("post").item( index: 0).getTextContent();

    //Adatok kiírása konzolra és a txt fájlba
    System.out.println("Player ID: " + idP);
    System.out.println("P_T FK: " + p_t);
    System.out.println("Name: " + name);
    System.out.println("Birthdate: " + birthdate);
    System.out.println("Post: " + post + "\n");
    fileWriter.write( str: "Player ID: " + idP + "\nP_T FK: " + p_t + "\nName: " + name + "\nBirthdate: " + birthdate + "\nPost: " + post + "\n\n");
}

}

System.out.println("-----\n");
fileWriter.write( str: "\n-----\n");

//Coach elem nodelistjének lekérdezése
nList = doc.getElementsByTagName("coach");

//Végig megyünk a listán
for (int i = 0; i < nList.getLength(); i++) {
    Node cNode = nList.item(i);

    System.out.println("Current element: " + cNode.getNodeName());
    if (cNode.getNodeType() == Node.ELEMENT_NODE) {

        // Elementé alakítjuk az aktuális elemet
        Element cElem = (Element) cNode;

        //Lekérjük az aktuális elem atributeumának és gyerekelemeinek tartalmát
        String idC = cElem.getAttribute( name: "idC");
        String c_t = cElem.getAttribute( name: "c_t");
        String name = cElem.getElementsByTagName("name").item( index: 0).getTextContent();
        NodeList nBirthdate = cElem.getElementsByTagName("birthdate");
        String birthdate = "";
        for (int j = 0; j < nBirthdate.getLength(); j++) {
            Node nDate = nBirthdate.item(j);
            if (nDate.getNodeType() == Node.ELEMENT_NODE) {
                Element eDate = (Element) nDate;
                birthdate = eDate.getElementsByTagName("year").item( index: 0).getTextContent();
                birthdate += "-" + eDate.getElementsByTagName("month").item( index: 0).getTextContent();
                birthdate += "-" + eDate.getElementsByTagName("day").item( index: 0).getTextContent();
            }
        }
        String licence = cElem.getElementsByTagName("licence").item( index: 0).getTextContent();

        //Adatok kiírása konzolra és a txt fájlba
        System.out.println("Coach ID: " + idC);
        System.out.println("C_T FK: " + c_t);
        System.out.println("Name: " + name);
        System.out.println("Birthdate: " + birthdate);
        System.out.println("Licence: " + licence + "\n");
        fileWriter.write( str: "Coach ID: " + idC + "\nC_T FK: " + c_t + "\nName: " + name + "\nBirthdate: " + birthdate + "\nLicence: " + licence + "\n\n");
    }
}

System.out.println("-----\n");
fileWriter.write( str: "\n-----\n");

//Match elem nodelistjének lekérdezése
nList = doc.getElementsByTagName("match");

//Végig megyünk a listán
for (int i = 0; i < nList.getLength(); i++) {
    Node mNode = nList.item(i);

    System.out.println("Current element: " + mNode.getNodeName());
    if (mNode.getNodeType() == Node.ELEMENT_NODE) {

        // Elementé alakítjuk az aktuális elemet
        Element mElem = (Element) mNode;

        //Lekérjük az aktuális elem atributeumának és gyerekelemeinek tartalmát
        String m_t = mElem.getAttribute( name: "m_t");
        String m_r = mElem.getAttribute( name: "m_r");
        String location = mElem.getElementsByTagName("location").item( index: 0).getTextContent();
        String date = mElem.getElementsByTagName("date").item( index: 0).getTextContent();

        //Adatok kiírása konzolra és a txt fájlba
        System.out.println("M_T FK: " + m_t);
        System.out.println("M_R FK: " + m_r);
        System.out.println("Location: " + location);
        System.out.println("Date: " + date + "\n");
        fileWriter.write( str: "M_T FK: " + m_t + "\nM_R FK: " + m_r + "\nLocation: " + location + "\nDate: " + date + "\n\n");
    }
}

```

```

229     }
230
231     System.out.println("-----\n");
232     fileWriter.write( str: "\n-----\n");
233
234     //Referee elem nodelistjének lekérdezése
235     nList = doc.getElementsByTagName("referee");
236
237     //Végig megyünk a listán
238     for (int i = 0; i < nList.getLength(); i++) {
239         Node rNode = nList.item(i);
240         System.out.println("Current element: " + rNode.getNodeName());
241         if (rNode.getNodeType() == Node.ELEMENT_NODE) {
242
243             // Elementé alakítjuk az aktuális elemet
244             Element rElem = (Element) rNode;
245
246             //Lekérjük az aktuális elem atributeumának és gyerekelemeinek tartalmát
247             String idR = rElem.getAttribute( name: "idR");
248             String name = rElem.getElementsByTagName("name").item( index: 0).getTextContent();
249             NodeList nBirthdate = rElem.getElementsByTagName("birthdate");
250             String birthdate = "";
251             for (int j = 0; j < nBirthdate.getLength(); j++) {
252                 Node nDate = nBirthdate.item(j);
253                 if (nDate.getNodeType() == Node.ELEMENT_NODE) {
254                     Element eDate = (Element) nDate;
255                     birthdate = eDate.getElementsByTagName("year").item( index: 0).getTextContent();
256                     birthdate += "-" + eDate.getElementsByTagName("month").item( index: 0).getTextContent();
257                     birthdate += "-" + eDate.getElementsByTagName("day").item( index: 0).getTextContent();
258                 }
259             }
260
261             //Adatok kiírása konzolra és a txt fájlba
262             String licence = rElem.getElementsByTagName("licence").item( index: 0).getTextContent();
263             System.out.println("Referee ID: " + idR);
264             System.out.println("Name: " + name);
265             System.out.println("Birthdate: " + birthdate);
266             System.out.println("Licence: " + licence + "\n");
267             fileWriter.write( str: "Referee ID: " + idR + "\nName: " + name + "\nBirthdate: " + birthdate + "\nLicence: " + licence + "\n\n");
268         }
269     }
270     fileWriter.flush();
271     fileWriter.close();
272 }
273

```



## 2.2 Adatlekérdezés

1. 40 évnél régebben alapított csapatok adatai!
2. Azon játékosok nevei, akik kapus poszton játszanak!
3. Teljes támogatási összeg csapatonként!
4. Mérkőzések helyszínei, amit egyes bírók fűjnek!
5. Mennyi NBII-es csapat van és melyek azok!

```
33 //40 évnél régebben alapított csapatok adatai
34 System.out.println("40 évnél régebben alapított csapatok adatai:\n");
35 NodeList teamList = doc.getElementsByTagName("team");
36 for (int i = 0; i < teamList.getLength(); i++) {
37     Node nNode = teamList.item(i);
38     if (nNode.getNodeType() == Node.ELEMENT_NODE) {
39         Element elem = (Element) nNode;
40         NodeList nlFoundation = elem.getElementsByTagName("foundation");
41         String foundation = "";
42         for (int j = 0; j < nlFoundation.getLength(); j++) {
43             Node nDate = nlFoundation.item(j);
44             if (nDate.getNodeType() == Node.ELEMENT_NODE) {
45                 Element eDate = (Element) nDate;
46                 if (Integer.parseInt(eDate.getElementsByTagName("year").item(0).getTextContent()) < 1982) {
47                     String idT = elem.getAttribute( "name": "idT");
48                     String name = elem.getElementsByTagName("name").item(0).getTextContent();
49                     foundation += eDate.getElementsByTagName("year").item(0).getTextContent();
50                     foundation += "-" + eDate.getElementsByTagName("month").item(0).getTextContent();
51                     foundation += "-" + eDate.getElementsByTagName("day").item(0).getTextContent();
52                     String rank = elem.getElementsByTagName("rank").item(0).getTextContent();
53                     System.out.println("Team ID: " + idT);
54                     System.out.println("Name: " + name);
55                     System.out.println("Foundation: " + foundation);
56                     System.out.println("Rank: " + rank);
57                     System.out.println();
58                 }
59             }
60         }
61     }
62 }
63 System.out.println("-----");
64
65 //Azon játékosok nevei, akik kapus poszton játszanak
66 System.out.println("Kapusok nevei!\n");
67 NodeList playerList = doc.getElementsByTagName("player");
68 for (int i = 0; i < playerList.getLength(); i++) {
69     Node nNode = playerList.item(i);
70     if (nNode.getNodeType() == Node.ELEMENT_NODE) {
71         Element elem = (Element) nNode;
72         if (elem.getElementsByTagName("post").item(0).getTextContent().equals("Kapus")) {
73             System.out.println("Név: " + elem.getElementsByTagName("name").item(0).getTextContent());
74         }
75     }
76 }
77 System.out.println("-----");
78
79 //Teljes támogatási összeg csapatonként
80 System.out.println("Csapatok összes támogatása:\n");
81 for (int i = 0; i < teamList.getLength(); i++) {
82     Node teamNode = teamList.item(i);
83     if (teamNode.getNodeType() == Node.ELEMENT_NODE) {
84         Element teamElem = (Element) teamNode;
85         String teamId = teamElem.getAttribute( "name": "idT");
86         int amount = 0;
87         NodeList supplist = doc.getElementsByTagName("supporter");
88         for (int j = 0; j < supplist.getLength(); j++) {
89             Node suppNode = supplist.item(j);
90             if (suppNode.getNodeType() == Node.ELEMENT_NODE) {
91                 Element suppElem = (Element) suppNode;
92                 String s_t = suppElem.getAttribute( "name": "s_t");
93                 if (teamId.equals(s_t)) {
94                     amount += Integer.parseInt(suppElem.getElementsByTagName("amount").item(0).getTextContent());
95                 }
96             }
97         }
98         System.out.println("A(z) " + teamElem.getElementsByTagName("name").item(0).getTextContent() + " csapat támogatása: " + amount + " Ft!");
99     }
100 }
101 System.out.println("-----");
```



```

103 //Mérkőzések helyszínei amit egyes bírók fűjnek
104 System.out.println("Bírók mérkőzéseinek helyszínei:\n");
105 NodeList refereeList = doc.getElementsByTagName("referee");
106 for (int i = 0; i < refereeList.getLength(); i++) {
107     Node rNode = refereeList.item(i);
108     if (rNode.getNodeType() == Node.ELEMENT_NODE) {
109         Element rElem = (Element) rNode;
110         String idR = rElem.getAttribute( "name": "idR");
111         NodeList matchList = doc.getElementsByTagName("match");
112         for (int j = 0; j < matchList.getLength(); j++) {
113             Node mNode = matchList.item(j);
114             if (mNode.getNodeType() == Node.ELEMENT_NODE) {
115                 Element mElem = (Element) mNode;
116                 String m_r = mElem.getAttribute( "name": "m_r");
117                 if (idR.equals(m_r)) {
118                     String name = rElem.getElementsByTagName("name").item( index: 0).getTextContent();
119                     String location = mElem.getElementsByTagName("location").item( index: 0).getTextContent();
120                     System.out.println(name + ": " + location);
121                 }
122             }
123         }
124     }
125 }
126 System.out.println("-----");
128 //Mennyi NBII-es csapat van és melyek azok?
129 System.out.println("Mennyi NBII-es csapat van és melyek azok?\n");
130 int counter = 0;
131 StringBuilder result = new StringBuilder();
132 for (int i = 0; i < teamList.getLength(); i++) {
133     Node tNode = teamList.item(i);
134     if (tNode.getNodeType() == Node.ELEMENT_NODE) {
135         Element tElem = (Element) tNode;
136         String rank = tElem.getElementsByTagName("rank").item( index: 0).getTextContent();
137         if ("NBII".equals(rank)) {
138             counter++;
139             result.append("-").append(tElem.getElementsByTagName("name").item( index: 0).getTextContent()).append("\n");
140         }
141     }
142 }

```

## 2.3 Adatmódosítás

1. Minden játékos posztját módosítjuk egy véletlenszerűen kiválasztottra!
2. Bírói licence vizsga: eggyel magasabb kategóriába kerül, ha a legmagasabban van akkor azt jelzi!
3. Támogatás meghosszabbítása 1 és 10 közötti évek számával!
4. Átigazolás 2 random játékosnál random csapathoz!
5. DVTK csapat rank módosítás!

```
30 //Minden játékos posztját módosítjuk egy véletlenszerűen kiválasztottra!
31 System.out.println("Postok módosítása minden játékosnál!\n");
32 NodeList playerList = doc.getElementsByTagName("player");
33 StringBuilder result = new StringBuilder();
34 int random;
35 for (int i = 0; i < playerList.getLength(); i++) {
36     Node pNode = playerList.item(i);
37     if (pNode.getNodeType() == Node.ELEMENT_NODE) {
38         Element pElem = (Element) pNode;
39         Node post = pElem.getElementsByTagName("post").item(0);
40         String name = pElem.getElementsByTagName("name").item(0).getTextContent();
41         String[] posts = {"Kapus", "Szélső", "Beálló", "Irányító", "Átlövő"};
42         random = (int) (Math.random() * posts.length);
43         result.append(name).append(" [eredeti post: ").append(post.getTextContent());
44         post.setTextContent(posts[random]);
45         result.append(", új post: ").append(post.getTextContent()).append("]\n");
46     }
47 }
48 System.out.print(result);
49 System.out.println("-----");
51 //Bírói licence vizsga: eggyel magasabb kategóriába kerül, ha a legmagasabban van akkor azt jelzi
52 System.out.println("Bírói licence vizsga!\n");
53 NodeList refereeList = doc.getElementsByTagName("referee");
54 for (int i = 0; i < refereeList.getLength(); i++) {
55     Node rNode = refereeList.item(i);
56     if (rNode.getNodeType() == Node.ELEMENT_NODE) {
57         Element elem = (Element) rNode;
58         Node licence = elem.getElementsByTagName("licence").item(0);
59         switch (licence.getTextContent()) {
60             case "A":
61                 System.out.println(elem.getElementsByTagName("name").item(0).getTextContent() + " már a legmagasabb licenccel rendelkezik!");
62                 break;
63             case "B":
64                 licence.setTextContent("A");
65                 System.out.println(elem.getElementsByTagName("name").item(0).getTextContent() + " " + licence.getTextContent() + " licenct kapott!");
66                 break;
67             case "C":
68                 licence.setTextContent("B");
69                 System.out.println(elem.getElementsByTagName("name").item(0).getTextContent() + " " + licence.getTextContent() + " licenct kapott!");
70                 break;
71             case "D":
72                 licence.setTextContent("C");
73                 System.out.println(elem.getElementsByTagName("name").item(0).getTextContent() + " " + licence.getTextContent() + " licenct kapott!");
74                 break;
75         }
76     }
77 }
78 System.out.println("-----");
```

```

80 //Támogatás meghosszabbítása 1 és 10 közötti évek számával
81 System.out.println("Minden támogató szerződés hosszabbítása!\n");
82 NodeList supporterList = doc.getElementsByTagName("supporter");
83 result = new StringBuilder();
84 for (int i = 0; i < supporterList.getLength(); i++) {
85     Node sNode = supporterList.item(i);
86     if (sNode.getNodeType() == Node.ELEMENT_NODE) {
87         Element sElem = (Element) sNode;
88         NodeList date = sElem.getElementsByTagName("validity");
89         result.append(sElem.getElementsByTagName("name").item( index: 0).getTextContent());
90         for (int j = 0; j < date.getLength(); j++) {
91             Node nDate = date.item(j);
92             if (nDate.getNodeType() == Node.ELEMENT_NODE) {
93                 Element dElem = (Element) nDate;
94                 Node year = dElem.getElementsByTagName("year").item( index: 0);
95                 String month = dElem.getElementsByTagName("month").item( index: 0).getTextContent();
96                 String day = dElem.getElementsByTagName("day").item( index: 0).getTextContent();
97                 result.append(" [előtte: ").append(year.getTextContent()).append("-").append(month).append("-").append(day);
98                 random = 1 + (int) (Math.random() * 11);
99                 year.setTextContent(String.valueOf( Integer.parseInt(year.getTextContent()) + random));
100                 result.append(", utánna: ").append(year.getTextContent()).append("-").append(month).append("-").append(day).append("]\n");
101             }
102         }
103     }
104 }
105 System.out.println(result);
106 System.out.println("-----");
107 //Átigazolás 2 random játékosnál random csapathoz
108 System.out.println("Két játékos átigazolása!\n");
109 String[] idTs = {"t1", "t2", "t3", "t4", "t5"};
110 result = new StringBuilder();
111 for (int i = 0; i < 2; i++) {
112     random = (int) (Math.random() * playerList.getLength());
113     Node pNode = playerList.item(random);
114     if (pNode.getNodeType() == Node.ELEMENT_NODE) {
115         Element pElem = (Element) pNode;
116         result.append(pElem.getElementsByTagName("name").item( index: 0).getTextContent()).append(" [előtte: ").append(pElem.getAttribute( name: "p_t"));
117         pElem.setAttribute( name: "p_t", idTs[(int) (Math.random() * idTs.length)]);
118         result.append(", utánna: ").append(pElem.getAttribute( name: "p_t")).append("]\n");
119     }
120 }
121 }
122 System.out.println(result);
123 System.out.println("-----");
124
125 //DVTK csapat rank módosítás
126 NodeList temaList = doc.getElementsByTagName("team");
127 result = new StringBuilder();
128 for (int i = 0; i < temaList.getLength(); i++) {
129     Node tNode = temaList.item(i);
130     if (tNode.getNodeType() == Node.ELEMENT_NODE) {
131         Element tElem = (Element) tNode;
132         Node rank = tElem.getElementsByTagName("rank").item( index: 0);
133         String name = tElem.getElementsByTagName("name").item( index: 0).getTextContent();
134         if ("DVTK".equals(name)){
135             result.append(name).append(" [előtte: ").append(rank.getTextContent());
136             rank.setTextContent("NBI/B");
137             result.append(", utánna: ").append(rank.getTextContent()).append("]\n");
138         }
139     }
140 }
141 System.out.println(result);

```

Torma Antal  
hallgató