

# JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Készítette: **Bajusz Máté Bence**

Neptunkód: **BMA015**

A feladatomban témája kosárlabda adatok nyilvántartása. Csapatokat, játékosokat, mérkőzéseket, játékos statisztikát tárolok.

A mérkőzéshez tartozik több játékos statisztika is, így itt N:M kapcsolat áll fenn. A játékos statisztika egy adott játékos egy adott mérkőzésen mért teljesítménye.

Maguk a játékosok és adataik a játékos statisztikában szerepelnek, itt is N:M kapcsolat áll fenn mivel egy játékosnak több meccsen, különböző ligákban elért adatait tárolhatjuk a játékos statisztikában.

## 1. feladat

```

    erDiagram
        Csapat ||--o{ Mérkőzés : "azonosít"
        Csapat ||--o{ Játékos_statisztika : "azonosít"
        Mérkőzés ||--o{ Játékos : "Össze-száml"
        Játékos ||--o{ Játékos_statisztika : "azonosít"

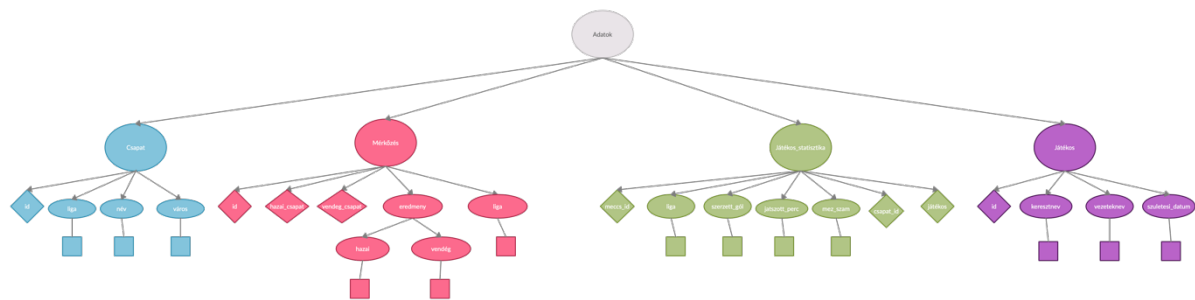
        Csapat {
            string név
            string id PK
            string város
            string liga
        }

        Mérkőzés {
            string dátum
            string id PK
            string hazai
            string vendég
            string eredmény
            string liga
            string vendég_csapat
            string hazai_csapat
        }

        Játékos {
            string id PK
            string keresztnév
            string vezeteknév
            string születési_datum
        }

        Játékos_statisztika {
            string meccs_id PK
            string mecszám
            string játékos
            string szerzett_gól
            string játszott_perc
            string csapat_id
            string liga
        }
  
```

### Az adatbázis konvertálása XDM modellre:



### c) Az XDM modell alapján XML dokumentum készítése:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><adatok
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="XMLSchemaBMA015.xsd">
  <jatekos id="j1">
    <keresztnev>Márton</keresztnev>
    <vezeteknev>Kozák</vezeteknev>
    <szulesesi_datum>1997.01.01</szulesesi_datum>
  </jatekos>
  <jatekos id="j2">
    <keresztnev>László</keresztnev>
    <vezeteknev>Polyák</vezeteknev>
    <szulesesi_datum>1997.01.01</szulesesi_datum>
  </jatekos>
  <jatekos id="j3">
    <keresztnev>Tamás</keresztnev>
    <vezeteknev>Kósa</vezeteknev>
    <szulesesi_datum>1999.01.01</szulesesi_datum>
  </jatekos>
  <jatekos id="j4">
    <keresztnev>János</keresztnev>
    <vezeteknev>Velkey</vezeteknev>
    <szulesesi_datum>1998.09.11</szulesesi_datum>
  </jatekos>
  <jatekos id="j5">
    <keresztnev>Bence</keresztnev>
    <vezeteknev>Tóth</vezeteknev>
    <szulesesi_datum>1997.08.15</szulesesi_datum>
  </jatekos>
  <jatekos id="j6">
    <keresztnev>Tibor</keresztnev>
    <vezeteknev>Bán</vezeteknev>
    <szulesesi_datum>1997.01.01</szulesesi_datum>
  </jatekos>
  <jatekos id="j7">
    <keresztnev>Csaba</keresztnev>
    <vezeteknev>Ferencz</vezeteknev>
    <szulesesi_datum>1997.12.25</szulesesi_datum>
  </jatekos>
  <jatekos id="j8">
    <keresztnev>Gergely</keresztnev>
    <vezeteknev>Lajsz</vezeteknev>
    <szulesesi_datum>1997.11.01</szulesesi_datum>
  </jatekos>
  <jatekos id="j9">
    <keresztnev>Bence</keresztnev>
    <vezeteknev>Oroszi</vezeteknev>
    <szulesesi_datum>1997.12.24</szulesesi_datum>
  </jatekos>
</adatok>
```

```
</jatekos>
<jatekos id="j10">
  <keresztnev>Milan</keresztnev>
  <vezeteknev>CSorvasi</vezeteknev>
  <szuletesi_datum>1998.02.02</szuletesi_datum>
</jatekos>
<jatekos id="j11">
  <keresztnev>Marcell</keresztnev>
  <vezeteknev>Érsek</vezeteknev>
  <szuletesi_datum>1997.03.15</szuletesi_datum>
</jatekos>
<jatekos id="j12">
  <keresztnev>Ákos</keresztnev>
  <vezeteknev>Németh</vezeteknev>
  <szuletesi_datum>1997.10.23</szuletesi_datum>
</jatekos>
<csapat id="cs1">
  <liga>hun</liga>
  <nev>DEAC</nev>
  <varos>Debrecen</varos>
</csapat>
<csapat id="cs2">
  <liga>hun</liga>
  <nev>Körmend</nev>
  <varos>Körmend</varos>
</csapat>
<csapat id="cs3">
  <liga>hun</liga>
  <nev>Pécsi VSK-VEOLIA</nev>
  <varos>Pécs</varos>
</csapat>
<csapat id="cs4">
  <liga>hun</liga>
  <nev>Alba Fehérvár</nev>
  <varos>Fehérvár</varos>
</csapat>
<merkozes hazai_csapat="cs1" id="mi1" vendeg_csapat="cs2">
  <eredmeny>
    <hazai>35</hazai>
    <vendeg>45</vendeg>
  </eredmeny>
  <liga>hun</liga>
</merkozes>
<merkozes hazai_csapat="cs3" id="mi2" vendeg_csapat="cs4">
  <eredmeny>
    <hazai>25</hazai>
    <vendeg>39</vendeg>
  </eredmeny>
  <liga>hun</liga>
</merkozes>
<jatekos_statisztika csapat_id="cs1" jatekos="j1" meccs_id="mi1">
  <liga>hun</liga>
  <szerzett_gol>5</szerzett_gol>
  <jatszott_perc>21</jatszott_perc>
  <mez_szam>11</mez_szam>
</jatekos_statisztika>
<jatekos_statisztika csapat_id="cs1" jatekos="j2" meccs_id="mi1">
  <liga>hun</liga>
  <szerzett_gol>4</szerzett_gol>
  <jatszott_perc>25</jatszott_perc>
  <mez_szam>22</mez_szam>
```

```
</jatekos_statisztika>
<jatekos_statisztika csapat_id="cs1" jatekos="j4" meccs_id="mi1">
  <liga>hun</liga>
  <szerzett_gol>3</szerzett_gol>
  <jatszott_perc>23</jatszott_perc>
  <mez_szam>13</mez_szam>
</jatekos_statisztika>
<jatekos_statisztika csapat_id="cs1" jatekos="j5" meccs_id="mi1">
  <liga>hun</liga>
  <szerzett_gol>13</szerzett_gol>
  <jatszott_perc>23</jatszott_perc>
  <mez_szam>99</mez_szam>
</jatekos_statisztika>
<jatekos_statisztika csapat_id="cs1" jatekos="j6" meccs_id="mi1">
  <liga>hun</liga>
  <szerzett_gol>8</szerzett_gol>
  <jatszott_perc>23</jatszott_perc>
  <mez_szam>89</mez_szam>
</jatekos_statisztika>
<jatekos_statisztika csapat_id="cs2" jatekos="j7" meccs_id="mi1">
  <liga>hun</liga>
  <szerzett_gol>7</szerzett_gol>
  <jatszott_perc>23</jatszott_perc>
  <mez_szam>29</mez_szam>
</jatekos_statisztika>
<jatekos_statisztika csapat_id="cs2" jatekos="j8" meccs_id="mi1">
  <liga>hun</liga>
  <szerzett_gol>11</szerzett_gol>
  <jatszott_perc>23</jatszott_perc>
  <mez_szam>39</mez_szam>
</jatekos_statisztika>
<jatekos_statisztika csapat_id="cs2" jatekos="j9" meccs_id="mi1">
  <liga>hun</liga>
  <szerzett_gol>6</szerzett_gol>
  <jatszott_perc>23</jatszott_perc>
  <mez_szam>44</mez_szam>
</jatekos_statisztika>
<jatekos_statisztika csapat_id="cs2" jatekos="j10" meccs_id="mi1">
  <liga>hun</liga>
  <szerzett_gol>12</szerzett_gol>
  <jatszott_perc>23</jatszott_perc>
  <mez_szam>56</mez_szam>
</jatekos_statisztika>
<jatekos_statisztika csapat_id="cs2" jatekos="j11" meccs_id="mi1">
  <liga>hun</liga>
  <szerzett_gol>4</szerzett_gol>
  <jatszott_perc>23</jatszott_perc>
  <mez_szam>66</mez_szam>
</jatekos_statisztika>
<jatekos_statisztika csapat_id="cs2" jatekos="j12" meccs_id="mi1">
  <liga>hun</liga>
  <szerzett_gol>5</szerzett_gol>
  <jatszott_perc>23</jatszott_perc>
  <mez_szam>73</mez_szam>
</jatekos_statisztika>
<jatekos_statisztika csapat_id="cs3" jatekos="j1" meccs_id="mi2">
  <liga>hun</liga>
  <szerzett_gol>6</szerzett_gol>
  <jatszott_perc>21</jatszott_perc>
  <mez_szam>11</mez_szam>
</jatekos_statisztika>
```

```
<jatekos_statisztika csapat_id="cs3" jatekos="j2" meccs_id="mi2">
  <liga>hun</liga>
  <szerzett_gol>7</szerzett_gol>
  <jatszott_perc>25</jatszott_perc>
  <mez_szam>22</mez_szam>
</jatekos_statisztika>
<jatekos_statisztika csapat_id="cs3" jatekos="j4" meccs_id="mi2">
  <liga>hun</liga>
  <szerzett_gol>2</szerzett_gol>
  <jatszott_perc>23</jatszott_perc>
  <mez_szam>13</mez_szam>
</jatekos_statisztika>
<jatekos_statisztika csapat_id="cs3" jatekos="j5" meccs_id="mi2">
  <liga>hun</liga>
  <szerzett_gol>9</szerzett_gol>
  <jatszott_perc>23</jatszott_perc>
  <mez_szam>99</mez_szam>
</jatekos_statisztika>
<jatekos_statisztika csapat_id="cs3" jatekos="j6" meccs_id="mi2">
  <liga>hun</liga>
  <szerzett_gol>1</szerzett_gol>
  <jatszott_perc>23</jatszott_perc>
  <mez_szam>89</mez_szam>
</jatekos_statisztika>
<jatekos_statisztika csapat_id="cs4" jatekos="j7" meccs_id="mi2">
  <liga>hun</liga>
  <szerzett_gol>3</szerzett_gol>
  <jatszott_perc>23</jatszott_perc>
  <mez_szam>29</mez_szam>
</jatekos_statisztika>
<jatekos_statisztika csapat_id="cs4" jatekos="j8" meccs_id="mi2">
  <liga>hun</liga>
  <szerzett_gol>3</szerzett_gol>
  <jatszott_perc>23</jatszott_perc>
  <mez_szam>39</mez_szam>
</jatekos_statisztika>
<jatekos_statisztika csapat_id="cs4" jatekos="j9" meccs_id="mi2">
  <liga>hun</liga>
  <szerzett_gol>4</szerzett_gol>
  <jatszott_perc>23</jatszott_perc>
  <mez_szam>44</mez_szam>
</jatekos_statisztika>
<jatekos_statisztika csapat_id="cs4" jatekos="j10" meccs_id="mi2">
  <liga>hun</liga>
  <szerzett_gol>5</szerzett_gol>
  <jatszott_perc>23</jatszott_perc>
  <mez_szam>56</mez_szam>
</jatekos_statisztika>
<jatekos_statisztika csapat_id="cs4" jatekos="j11" meccs_id="mi2">
  <liga>hun</liga>
  <szerzett_gol>5</szerzett_gol>
  <jatszott_perc>23</jatszott_perc>
  <mez_szam>66</mez_szam>
</jatekos_statisztika>
<jatekos_statisztika csapat_id="cs4" jatekos="j12" meccs_id="mi2">
  <liga>hun</liga>
  <szerzett_gol>5</szerzett_gol>
  <jatszott_perc>23</jatszott_perc>
  <mez_szam>73</mez_szam>
</jatekos_statisztika>
</adatok>
```

## 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="adatok">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="jatekos"/>
        <xs:element maxOccurs="unbounded" ref="csapat"/>
        <xs:element maxOccurs="unbounded" ref="merkozes"/>
        <xs:element maxOccurs="unbounded"
ref="jatekos_statisztika"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="jatekos">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="keresztnev"/>
        <xs:element ref="vezeteknev"/>
        <xs:element ref="szuletesi_datum"/>
      </xs:sequence>
      <xs:attribute name="id" use="required" type="xs:ID" />
    </xs:complexType>
  </xs:element>
  <xs:element name="keresztnev" type="xs:string"/>
  <xs:element name="vezeteknev" type="xs:string"/>
  <xs:element name="szuletesi_datum" type="xs:string"/>
  <xs:element name="csapat">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="liga"/>
        <xs:element ref="nev"/>
        <xs:element ref="varos"/>
      </xs:sequence>
      <xs:attribute name="id" use="required" type="xs:ID"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="nev" type="xs:string"/>
  <xs:element name="varos" type="xs:string"/>
  <xs:element name="merkozes">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="eredmeny"/>
        <xs:element ref="liga"/>
      </xs:sequence>
      <xs:attribute name="hazai_csapat" use="required"
type="xs:string"/>
      <xs:attribute name="id" use="required" type="xs:ID"/>
      <xs:attribute name="vendeg_csapat" use="required"
type="xs:string"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="eredmeny">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="hazai"/>

```

```

        <xs:element ref="vendeg"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="hazai" type="xs:integer"/>
<xs:element name="vendeg" type="xs:integer"/>
<xs:element name="jatekos_statisztika">
    <xs:complexType>
        <xs:sequence >
            <xs:element name="liga"/>
            <xs:element name="szerzett_gol"/>
            <xs:element name="jatszott_perc"/>
            <xs:element ref="mez_szam"/>
        </xs:sequence>
        <xs:attribute name="csapat_id" type="xs:string"/>
        <xs:attribute name="jatekos" use="required" type="xs:string"/>
        <xs:attribute name="meccs_id" use="required" type="xs:string"/>
    </xs:complexType>
</xs:element>
<xs:element name="jatszott_perc" type="xs:integer"/>
<xs:element name="szerzett_gol" type="xs:integer"/>
<xs:element name="mez_szam" type="xs:integer"/>
<xs:element name="liga" type="xs:string"/>
</xs:schema>

```

## 2. feladat

### 2a) adatolvasás

```

package hu.domparsa.bma015;

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

public class DOMReadBMA015 {

    //összes játékos
    public void GetPlayers(Document doc) {
        System.out.println("\n játékosok:");
        NodeList nodeList = doc.getElementsByTagName("jatekos");
        for (int i = 0; i < nodeList.getLength(); i++) {
            Node node = nodeList.item(i);
            System.out.println();
            if (node.getNodeType() == Node.ELEMENT_NODE) {
                Element element = (Element) node;
                System.out.println("játékos azonosító: " +
                    element.getAttribute("id"));
                System.out.println("vezetéknev: " +
                    element.getElementsByTagName("vezeteknev").item(0).getTextContent());
                System.out.println("keresztnev: " +
                    element.getElementsByTagName("keresztnev").item(0).getTextContent());
                System.out.println("születési dátum: " +
                    element.getElementsByTagName("szuletesi_datum").item(0).getTextContent());
            }
        }
    }
}

```



```

//összes csapat
public void GetTeams(Document doc) {
    System.out.println("\n csapatok:");
    NodeList nodeList = doc.getElementsByTagName("csapat");
    for (int i = 0; i < nodeList.getLength(); i++) {
        Node node = nodeList.item(i);
        System.out.println();
        if (node.getNodeType() == Node.ELEMENT_NODE) {
            Element element = (Element) node;
            System.out.println("csapat azonosító: " +
element.getAttribute("id"));
            System.out.println("bajnokság / liga: " +
element.getElementsByTagName("liga").item(0).getTextContent());
            System.out.println("Csapat név: " +
element.getElementsByTagName("nev").item(0).getTextContent());
            System.out.println("város: " +
element.getElementsByTagName("varos").item(0).getTextContent());
        }
    }
}

//összes merkozes, és a hozzá tartozó csapat nevek, játékos
statisztikák
public void GetGames(Document doc) {
    System.out.println("\n mérkőzés információk:");
    NodeList merkozesek = doc.getElementsByTagName("merkozes");
    //
    NodeList csapatok = doc.getElementsByTagName("csapat");
    for (int i = 0; i < merkozesek.getLength(); i++) {
        Node node = merkozesek.item(i);
        System.out.println();
        if (node.getNodeType() == Node.ELEMENT_NODE) {
            Element element = (Element) node;
            System.out.println("mérkőzés azonosító: " +
element.getAttribute("id"));
            System.out.println("bajnokság / liga: " +
element.getElementsByTagName("liga").item(0).getTextContent());
            System.out.println("hazai csapat: " + getTagValueByID(doc,
element.getAttribute("hazai_csapat"), "nev"));
            System.out.println("vendég csapat: " +
getTagValueByID(doc, element.getAttribute("vendeg_csapat"), "nev"));
            Element eredmeny = (Element)
element.getElementsByTagName("eredmeny").item(0);
            System.out.println("hazai eredmény: "
+eredmeny.getElementsByTagName("hazai").item(0).getTextContent());
            System.out.println("vendég eredmény: " +
eredmeny.getElementsByTagName("vendeg").item(0).getTextContent());
        }
    }
}

//Az összes játékos statisztika, játékosonként csoportosítva
public void GetAllPlayerStatisticsByPlayer(Document doc) {
    System.out.println("\n minden játékos statisztikája, játékosonként
csoportosítva");
    NodeList jatekosok = doc.getElementsByTagName("jatekos");
    NodeList statistics =
doc.getElementsByTagName("jatekos_statisztika");
    for (int i = 0; i < jatekosok.getLength(); i++) {
        Node jatekos = jatekosok.item(i);
        if (jatekos.getNodeType() == Node.ELEMENT_NODE) {
            Element element = (Element) jatekos;
            String id = element.getAttribute("id");

```

```

System.out.println("*****");
    System.out.println("játékos azonosító: " + id);
    System.out.println("vezetéknév: " +
element.getElementsByTagName("vezeteknev").item(0).getTextContent());
    System.out.println("keresztnev: " +
element.getElementsByTagName("keresztnev").item(0).getTextContent());
    System.out.println("születési dátum: " +
element.getElementsByTagName("szuletesi_datum").item(0).getTextContent());
    //ez lekérdezi az adott játékoshoz tartozó statisztikákat
    for (int j = 0; j < statistics.getLength(); j++) {
        Node stat = statistics.item(j);
        if (stat.getNodeType() == Node.ELEMENT_NODE) {
            Element statElement = (Element) stat;
            if
(statElement.getAttribute("jatekos").equals(id)) {
                System.out.println("mérkőzés azonosító: " +
statElement.getAttribute("meccs_id"));
                System.out.println("csapatban játszott: " +
getTagValueByID(doc, statElement.getAttribute("csapat_id"), "nev"));
                System.out.println("liga: " +
statElement.getElementsByTagName("liga").item(0).getTextContent());
                System.out.println("szerzett gol: " +
statElement.getElementsByTagName("szerzett_gol").item(0).getTextContent());
                System.out.println("jatszott_perc: " +
statElement.getElementsByTagName("jatszott_perc").item(0).getTextContent());
            }
        }
    }
}
//ez egy id alapján megkeresi az element-et majd a megfelelő tag
értékét vissza adja
//Csak akkor működik ha schema fájlban megfelelően van definiálva az id
public String getTagValueByID(Document doc, String id, String tag) {
    Element element = doc.getElementById(id);
    return element.getElementsByTagName(tag).item(0).getTextContent();
}
}

```

## 2b) adatmódosítás

```

package hu.domparse.bma015;

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

import java.util.Scanner;

public class DomModifyBMA015 {

    public void ModifyTeam(Document doc) {
        Scanner scan = new Scanner(System.in);
        System.out.println("Adja meg a modositandó csapat id-jét");
        String id = scan.nextLine();
        try {
            Element csapat = doc.getElementById(id);

```

```

        Node liga = csapat.getElementsByTagName("liga").item(0);
        Node nev = csapat.getElementsByTagName("nev").item(0);
        Node varos = csapat.getElementsByTagName("varos").item(0);
        System.out.println("Írja be az új ligát");
        String l = scan.nextLine();
        System.out.println("Írja be az új nevet");
        String n = scan.nextLine();
        System.out.println("Írja be az új várost");
        String v = scan.nextLine();

        liga.setTextContent(l);
        nev.setTextContent(n);
        varos.setTextContent(v);

        System.out.println("Módosítás sikeres");
        System.out.println("csapat azonosító: " +
csapat.getAttribute("id"));
        System.out.println("bajnokság / liga: " +
csapat.getElementsByTagName("liga").item(0).getTextContent());
        System.out.println("Csapat név: " +
csapat.getElementsByTagName("nev").item(0).getTextContent());
        System.out.println("város: " +
csapat.getElementsByTagName("varos").item(0).getTextContent());
    } catch (Exception exception) {
        System.out.println("nem sikerült a módosítás");
    }
}

public void ModifyPlayer(Document doc) {
    Scanner scan = new Scanner(System.in);
    System.out.println("Adj meg a módosítandó játékos id-jét");
    String id = scan.nextLine();
    try {
        Element jatekos = doc.getElementById(id);
        Node knev = jatekos.getElementsByTagName("keresztnev").item(0);
        Node vnev = jatekos.getElementsByTagName("vezeteknev").item(0);
        Node szdatum =
jatekos.getElementsByTagName("szuletesi_datum").item(0);
        System.out.println("Írja be az új keresztnévet");
        String k = scan.nextLine();
        System.out.println("Írja be az új vezetéknévet");
        String v = scan.nextLine();
        System.out.println("Írja be az új születési dátumot");
        String sz = scan.nextLine();
        knev.setTextContent(k);
        vnev.setTextContent(v);
        szdatum.setTextContent(sz);
        System.out.println("Módosítás sikeres");
        System.out.println("játékos azonosító: " +
jatekos.getAttribute("id"));
        System.out.println("keresztnév: " +
jatekos.getElementsByTagName("keresztnev").item(0).getTextContent());
        System.out.println("vezetéknév: " +
jatekos.getElementsByTagName("vezeteknev").item(0).getTextContent());
        System.out.println("születési dátum: " +
jatekos.getElementsByTagName("szuletesi_datum").item(0).getTextContent());
    } catch (Exception exception) {
        System.out.println("nem sikerült a módosítás");
    }
}

//Módosítja a játékos mérkőzést

```

```

public void ModifyGame(Document doc) {
    Scanner scan = new Scanner(System.in);
    System.out.println("Adja meg a módosítandó mérkőzés id-jét");
    String id = scan.nextLine();
    try {
        Element merkozes = doc.getElementById(id);
        Node liga = merkozes.getElementsByTagName("liga").item(0);
        Element eredmeny = (Element)
merkozes.getElementsByTagName("eredmeny").item(0);
        System.out.println("Írja be az új hazai csapat ID-jét");
        String hazaiID = scan.nextLine();
        System.out.println("Írja be az új vendég csapat ID-jét");
        String vendegID = scan.nextLine();
        System.out.println("Írja be az új eredményt hazai részről");
        String hEredmeny = scan.nextLine();
        System.out.println("Írja be az új eredményt vendég részről");
        String vEredmeny = scan.nextLine();
        System.out.println("Írja be az új ligát");
        String l = scan.next();

        liga.setTextContent(l);
        merkozes.setAttribute("hazai_csapat", hazaiID);
        merkozes.setAttribute("vendeg_csapat", vendegID);

eredmeny.getElementsByTagName("hazai").item(0).setTextContent(hEredmeny);
eredmeny.getElementsByTagName("vendeg").item(0).setTextContent(vEredmeny);

        System.out.println("Módosítás sikeres");
        System.out.println("mérkőzés azonosító: " + id);
        System.out.println("bajnokság / liga: " +
liga.getTextContent());
        //ugye itt a csapat neveket nem tartalmazza a merkozes, csak az
attributum utal a csapatra egy id-vel,
        // ezért a nevüket le kell kérdezni a megfelelő csapat element-
ből
        System.out.println("hazai csapat: " + getTagValueByID(doc,
merkozes.getAttribute("hazai_csapat"), "nev"));
        System.out.println("vendég csapat: " + getTagValueByID(doc,
merkozes.getAttribute("vendeg_csapat"), "nev"));
        //
        Element eredmeny = (Element)
merkozes.getElementsByTagName("eredmeny").item(0);
        System.out.println("hazai eredmény: "
+eredmeny.getElementsByTagName("hazai").item(0).getTextContent());
        System.out.println("vendég eredmény: " +
eredmeny.getElementsByTagName("vendeg").item(0).getTextContent());

    } catch (Exception exception) {
        System.out.println("nem sikerült a módosítás");
    }

}

//Módosítja a játékos statisztikát
public void ModifyPlayerStatistic(Document doc) {
    Scanner scan = new Scanner(System.in);
    System.out.println("Adja meg a módosítandó játékos statisztika
játékos id-jét");
    String jatekosID = scan.nextLine();
    System.out.println("Adja meg a módosítandó játékos statisztika
meccs id-jét");

```

```

        String meccsID = scan.nextLine();
        NodeList statistics =
doc.getElementsByTagName("jatekos_statisztika");
        try {
            for (int i = 0; i < statistics.getLength(); i++) {
                Node stat = statistics.item(i);
                if (stat.getNodeType() == Node.ELEMENT_NODE) {
                    Element statElement = (Element) stat;
                    if
(statElement.getAttribute("jatekos").equals(jatekosID) &&
statElement.getAttribute("meccs_id").equals(meccsID)){
                        System.out.println("Írja be az új ligát");
                        String l = scan.nextLine();
                        System.out.println("Írja be az szerzett gólokat");
                        String gol = scan.nextLine();
                        System.out.println("Írja be az új játszott
perceket");

                        String perc = scan.nextLine();
                        System.out.println("Írja be az új mez számot");
                        String mez = scan.nextLine();

statElement.getElementsByTagName("liga").item(0).setTextContent(l);

statElement.getElementsByTagName("szerzett_gol").item(0).setTextContent(gol
);

statElement.getElementsByTagName("jatszott_perc").item(0).setTextContent(pe
rc);

statElement.getElementsByTagName("mez_szam").item(0).setTextContent(mez);
                        System.out.println("Módosítás sikeres\n");

                        System.out.println("liga: " +
statElement.getElementsByTagName("liga").item(0).getTextContent());
                        System.out.println("szerzett_gol: " +
statElement.getElementsByTagName("szerzett_gol").item(0).getTextContent());
                        System.out.println("jatszott_perc: " +
statElement.getElementsByTagName("jatszott_perc").item(0).getTextContent())
;

                        System.out.println("mez: " +
statElement.getElementsByTagName("mez_szam").item(0).getTextContent());

                    }
                }
            }

        } catch (Exception exception) {
            System.out.println("nem sikerült a módosítás");
        }
    }
    //ez egy id alapján megkeresi az element-et majd a megfelelő tag
értékét vissza adja
    //Csak akkor működik ha schema fájlban megfelelően van definiálva az id
    public String getTagValueByID(Document doc, String id, String tag) {
        Element element = doc.getElementById(id);
        return element.getElementsByTagName(tag).item(0).getTextContent();
    }
}

```

