

# Jegyzőkönyv

Adatkezelés XML környezetben

Féléves feladat

Labdarúgó bajnokság

Készítette: Nyeste Ágoston

Neptunkód: WH85ZH

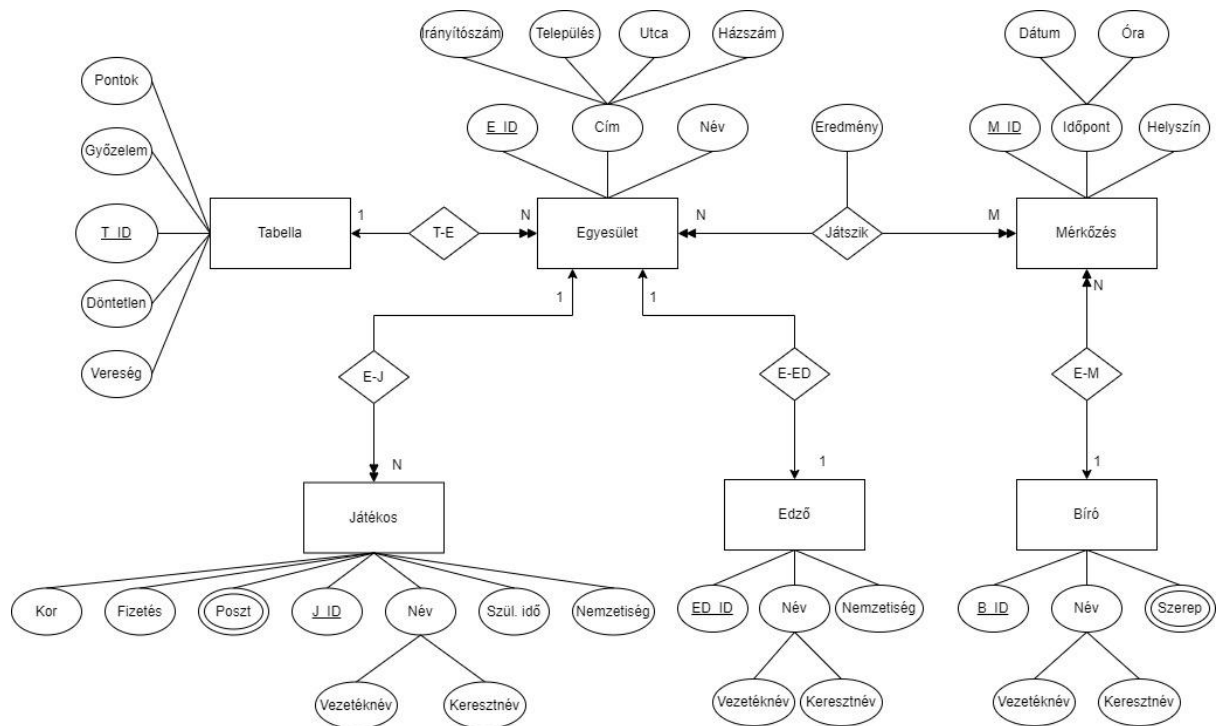
A feladathoz egy labdarúgó bajnokság szezon végi adatbázis rendszerét választottam.

**Egyedek:** Tabella, Egyesület, Játékos, Mérkőzés, Bíró, Edző

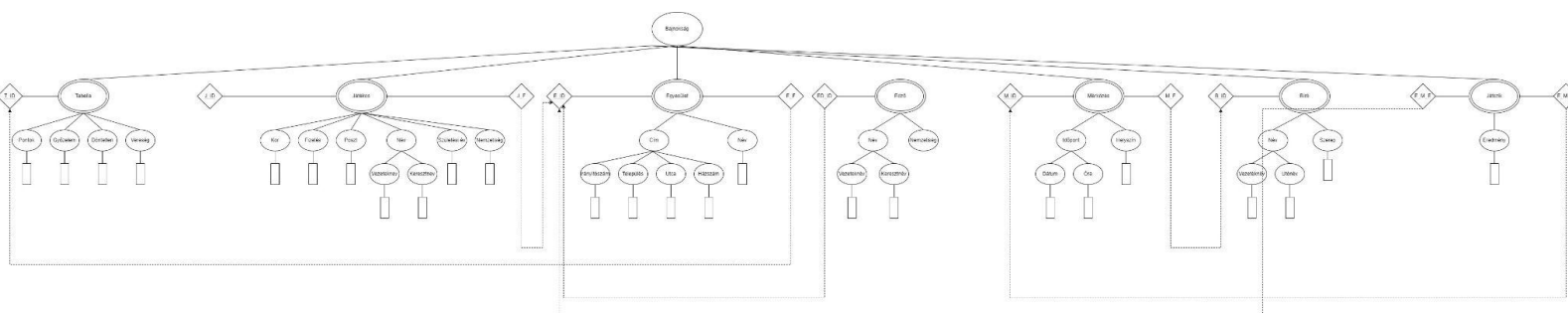
**Kapcsolatok:**

- Tabella - Egyesület: 1-N kapcsolat, mert egy egyesület egyszer szerepelhet a tabellán, a tabellán viszont több Egyesület is szerepel (az összes egyesület szerepel a tabellán).
- Egyesület – Játékos: 1-N kapcsolat, mivel egy játékos egyszerre csak egy egyesület tagja lehet, viszont egy egyesülethez több játékos is tartozik.
- Mérkőzés – Egyesület: N-M kapcsolat, egy mérkőzéshez több egyesület tartozik (pontosan 2), egy egyesület több mérkőzést is játszik az idény során. Neve: Játsszik saját adattagja a Mikor.
- Bíró - Mérkőzés: 1-N kapcsolat, mert egy Bíró több különböző mérkőzésen is fűjhat/dolgozhat, de egy adott mérkőzésen egy Bíró csak egyszer szerepelhet.
- Egyesület - Edző: 1-1 kapcsolat, mivel egy egyesületnek egyszerre egy edzője lehet és egy edző egy időben egy csapathoz tartozik.

## 1a) Az adatbázis ER modell készítése:



## 1b) Az ER modell konvertálása XDM modellre



## 1c) Az XDM modell alapján validált XML dokumentum készítése

```
<?xml version="1.0" encoding="utf-8" ?>

<bajnoksag xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <tabella T_ID="1">
    <pontok>18</pontok>
    <gyozelem>5</gyozelem>
    <dontetlen>3</dontetlen>
    <veresege>2</veresege>
  </tabella>
  <tabella T_ID="2">
    <pontok>15</pontok>
    <gyozelem>3</gyozelem>
    <dontetlen>6</dontetlen>
    <veresege>1</veresege>
  </tabella>
  <tabella T_ID="3">
    <pontok>14</pontok>
    <gyozelem>3</gyozelem>
    <dontetlen>5</dontetlen>
    <veresege>2</veresege>
  </tabella>

  <jatekos J_ID="1" J_F="1">
    <kor>31</kor>
    <fizetes>100</fizetes>
    <poszt>kozepallyas</poszt>
    <vezeteknev>Kanté</vezeteknev>
    <keresztnev>N Golo</keresztnev>
    <szulev>1991</szulev>
    <nemzetiseg>Francia</nemzetiseg>
  </jatekos>
```

```
<jatekos J_ID="2" J_F="2">
  <kor>34</kor>
  <fizetes>120</fizetes>
  <poszt>tamado</poszt>
  <vezeteknev>Lewandowski</vezeteknev>
  <keresztnev>Kanté</keresztnev>
  <szulev>1988</szulev>
  <nemzetiseg>Lengyel</nemzetiseg>
</jatekos>
<jatekos J_ID="3" J_F="3">
  <kor>22</kor>
  <fizetes>90</fizetes>
  <poszt>kozeppalyas</poszt>
  <vezeteknev>Haaland</vezeteknev>
  <keresztnev>Erling</keresztnev>
  <szulev>2000</szulev>
  <nemzetiseg>Norvég</nemzetiseg>
</jatekos>

<egyesulet E_ID="1" E_F="1">
  <iranyitoszam>1012</iranyitoszam>
  <telepules>London</telepules>
  <utca>Stadium</utca>
  <hazszam>1</hazszam>
  <nev>Chelsea</nev>
</egyesulet>
<egyesulet E_ID="2" E_F="2">
  <iranyitoszam>1012</iranyitoszam>
  <telepules>Barcelona</telepules>
  <utca>Senour</utca>
  <hazszam>33</hazszam>
  <nev>Barcelona</nev>
</egyesulet>
```

```
<egyesulet E_ID="3" E_F="3">
  <iranyitoszam>1012</iranyitoszam>
  <telepules>Manchester</telepules>
  <utca>Garden</utca>
  <hazszam>10</hazszam>
  <nev>Manchester City</nev>
</egyesulet>
```

```
<merkozes M_ID="1" M_F="1">
  <datum>2022.10.12</datum>
  <ora>21:00</ora>
  <helyszin>London</helyszin>
</merkozes>
```

```
<merkozes M_ID="2" M_F="2">
  <datum>2022.08.05</datum>
  <ora>18:00</ora>
  <helyszin>Barcelona</helyszin>
</merkozes>
```

```
<merkozes M_ID="3" M_F="3">
  <datum>2022.06.28</datum>
  <ora>20:00</ora>
  <helyszin>Paris</helyszin>
</merkozes>
```

```
<biro B_ID="1">
  <szerep>Pálya</szerep>
  <vezeteknev>Collina</vezeteknev>
  <keresztnev>Pierluigi</keresztnev>
</biro>
```

```
<biro B_ID="2">
  <szerep>Pálya</szerep>
  <vezeteknev>Kassai</vezeteknev>
  <keresztnev>Viktor</keresztnev>
</biro>
```

```
<biro B_ID="3">
  <szerep>Partjelző</szerep>
  <vezeteknev>Pedro</vezeteknev>
  <keresztnev>Poenca</keresztnev>
</biro>
```

```
<edzo ED_ID="1">
  <vezeteknev>Tuchel</vezeteknev>
  <keresztnev>Thomsa</keresztnev>
  <nemzetiseg>Német</nemzetiseg>
</edzo>
```

```
<edzo ED_ID="2">
  <vezeteknev>Alonso</vezeteknev>
  <keresztnev>Xavi</keresztnev>
  <nemzetiseg>Spanyol</nemzetiseg>
</edzo>
```

```
<edzo ED_ID="3">
  <vezeteknev>Guardiola</vezeteknev>
  <keresztnev>Pep</keresztnev>
  <nemzetiseg>Spanyol</nemzetiseg>
</edzo>
```

S

```
<jatszik E_M_E="1" E_M_M="1">
  <eredmeny>2</eredmeny>
</jatszik>
<jatszik E_M_E="2" E_M_M="1">
  <eredmeny>1</eredmeny>
</jatszik>
<jatszik E_M_E="3" E_M_M="3">
  <eredmeny>1</eredmeny>
</jatszik>
```

```
</bajnoksag>
```

## 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" attributeFormDefault="qualified">

  <!-- Elements, types -->

  <xs:element name="pontok" type="xs:integer" />
  <xs:element name="gyozelem" type="xs:integer" />
  <xs:element name="dontetlen" type="xs:integer" />
  <xs:element name="vereseg" type="xs:integer" />
  <xs:element name="kor" type="xs:string" />
  <xs:element name="fizetes" type="xs:integer" />
  <xs:element name="poszt" type="xs:string" />
  <xs:element name="vezeteknev" type="xs:string" />
  <xs:element name="keresztnev" type="xs:string" />
  <xs:element name="szulev" type="xs:integer" />
  <xs:element name="nemzetiseg" type="xs:string" />
  <xs:element name="iranyitoszam" type="xs:integer" />
  <xs:element name="telepules" type="xs:string" />
  <xs:element name="utca" type="xs:string" />
  <xs:element name="hazszam" type="xs:integer" />
  <xs:element name="nev" type="xs:string" />
  <xs:element name="datum" type="xs:string" />
  <xs:element name="ora" type="xs:string" />
  <xs:element name="helyszin" type="xs:string" />
  <xs:element name="szerep" type="xs:string" />
  <xs:element name="eredmeny" type="xs:integer" />

</xs:schema>
```



```

<xs:attribute name="T_ID" type="xs:integer" />
<xs:attribute name="J_ID" type="xs:integer" />
<xs:attribute name="J_F" type="xs:integer" />
<xs:attribute name="E_ID" type="xs:integer" />
<xs:attribute name="E_F" type="xs:integer" />
<xs:attribute name="M_ID" type="xs:integer" />
<xs:attribute name="M_F" type="xs:integer" />
<xs:attribute name="B_ID" type="xs:integer" />
<xs:attribute name="ED_ID" type="xs:integer" />
<xs:attribute name="E_M_E" type="xs:integer" />
<xs:attribute name="E_M_M" type="xs:integer" />

<!-- Complex types -->

<xs:complexType name="tabella_tipus">
  <xs:sequence>
    <xs:element ref="pontok"/>
    <xs:element ref="gyozelem"/>
    <xs:element ref="dontetlen"/>
    <xs:element ref="veresege"/>
  </xs:sequence>
  <xs:attribute ref="T_ID" use="required"/>
</xs:complexType>

<xs:complexType name="jatekos_tipus">
  <xs:sequence>
    <xs:element ref="kor"/>
    <xs:element ref="fizetes"/>
    <xs:element ref="poszt"/>
    <xs:element ref="vezeteknev"/>
    <xs:element ref="keresztnev"/>
    <xs:element ref="szulev"/>
    <xs:element ref="nemzetiseg"/>
  </xs:sequence>
  <xs:attribute ref="J_ID" use="required"/>
  <xs:attribute ref="J_F" use="required"/>

```

```
<xs:complexType name="egyesulet_tipus">
  <xs:sequence>
    <xs:element ref="iranyitoszam"/>
    <xs:element ref="telepules"/>
    <xs:element ref="utca"/>
    <xs:element ref="hazszam"/>
    <xs:element ref="nev"/>
  </xs:sequence>
  <xs:attribute ref="E_ID" use="required"/>
  <xs:attribute ref="E_F" use="required"/>
</xs:complexType>

<xs:complexType name="merkozes_tipus">
  <xs:sequence>
    <xs:element ref="datum"/>
    <xs:element ref="ora"/>
    <xs:element ref="helyszin"/>
  </xs:sequence>
  <xs:attribute ref="M_ID" use="required"/>
  <xs:attribute ref="M_F" use="required"/>
</xs:complexType>

<xs:complexType name="biro_tipus">
  <xs:sequence>
    <xs:element ref="szerep"/>
    <xs:element ref="vezeteknev"/>
    <xs:element ref="keresztnev"/>
  </xs:sequence>
  <xs:attribute ref="B_ID" use="required"/>
</xs:complexType>

<xs:complexType name="edzo_tipus">
  <xs:sequence>
    <xs:element ref="vezeteknev"/>
    <xs:element ref="keresztnev"/>
    <xs:element ref="nemzetiseg"/>
  </xs:sequence>
</xs:complexType>
```

```

<xs:complexType name="jatszik_tipus">
  <xs:sequence>
    <xs:element ref="eredmeny"/>
  </xs:sequence>
  <xs:attribute ref="E_M_E" use="required"/>
  <xs:attribute ref="E_M_M" use="required"/>
</xs:complexType>

<xs:element name="bajnoksag">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="tabella" type="tabella_tipus" maxOccurs="unbounded"/>
      <xs:element name="jatekos" type="jatekos_tipus" maxOccurs="unbounded"/>
      <xs:element name="egyesulet" type="egyesulet_tipus" maxOccurs="unbounded"/>
      <xs:element name="merkozes" type="merkozes_tipus" maxOccurs="unbounded"/>
      <xs:element name="biro" type="biro_tipus" maxOccurs="unbounded"/>
      <xs:element name="edzo" type="edzo_tipus" maxOccurs="unbounded"/>
      <xs:element name="jatszik" type="jatszik_tipus" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <!-- Keys -->

  <xs:key name="T_ID">
    <xs:selector xpath="tabella"/>
    <xs:field xpath="@T_ID"/>
  </xs:key>

  <xs:key name="J_ID">
    <xs:selector xpath="jatekos"/>
    <xs:field xpath="@J_ID"/>
  </xs:key>

  <xs:key name="E_ID">
    <xs:selector xpath="egyesulet"/>
    <xs:field xpath="@E_ID"/>
  </xs:key>

```

```

<xs:key name="E_ID">
  <xs:selector xpath="egyesulet"/>
  <xs:field xpath="@E_ID"/>
</xs:key>

<xs:key name="M_ID">
  <xs:selector xpath="merkozes"/>
  <xs:field xpath="@M_ID"/>
</xs:key>

<xs:key name="B_ID">
  <xs:selector xpath="biro"/>
  <xs:field xpath="@B_ID"/>
</xs:key>

<xs:key name="ED_ID">
  <xs:selector xpath="edzo"/>
  <xs:field xpath="@ED_ID"/>
</xs:key>

<!-- Foreign keys -->

<xs:keyref name="egyesulet_FK" refer="E_ID">
  <xs:selector xpath="jatekos"></xs:selector>
  <xs:field xpath="@J_F"></xs:field>
</xs:keyref>

<xs:keyref name="tabella_FK" refer="E_ID">
  <xs:selector xpath="egyesulet"></xs:selector>
  <xs:field xpath="@E_F"></xs:field>
</xs:keyref>

<xs:keyref name="biro_FK" refer="B_ID">
  <xs:selector xpath="merkozes"></xs:selector>
  <xs:field xpath="@M_F"></xs:field>
</xs:keyref>

```

```

    <xs:keyref name="jatszikk_egyesulet_FK" refer="E_ID">
      <xs:selector xpath="jatszikk"></xs:selector>
      <xs:field xpath="@E_M_E"></xs:field>
    </xs:keyref>

    <xs:keyref name="jatszikk_merkozes_FK" refer="M_ID">
      <xs:selector xpath="jatszikk"></xs:selector>
      <xs:field xpath="@E_M_M"></xs:field>
    </xs:keyref>
  </xs:element>
</xs:schema>

```

## 2.Feladat Adatolvasás

```
package hu.domparsing.wh85zh;

import java.io.File;
import java.io.IOException;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;

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

public class DOMReadWH85ZH {

    Run | Debug
    public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException {
        //Forrás file
        File file = new File(pathname: "XMLWH85ZH.xml");

        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = factory.newDocumentBuilder();

        Document doc = dBuilder.parse(file);

        doc.getDocumentElement().normalize();
        //Gyökérelem
        System.out.println("Root element: " + doc.getDocumentElement().getNodeName());
        //Gyerekelemek lementése
        NodeList nList = (NodeList) doc.getDocumentElement().getElementsByTagName("item");

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

```

//Ha jatekoshoz erünk
if(node.getNodeName() == "jatekos"){
    if(!node.getNodeName().equals(anObject: "#text")) {
        System.out.println(x: "\n");
        System.out.println("Current element: " + node.getNodeName());
    }
    //Játékosok adatainak kiírása
    if(node.getNodeType() == Node.ELEMENT_NODE) {
        Element elem = (Element) node;

        String uid = elem.getAttribute(name: "J_ID");

        Node node1 = elem.getElementsByTagName(name: "vezeteknev").item(index: 0);
        String fname = node1.getTextContent();

        Node node2 = elem.getElementsByTagName(name: "keresztnev").item(index: 0);
        String lname = node2.getTextContent();

        Node node3 = elem.getElementsByTagName(name: "nemzetiseg").item(index: 0);
        String nation = node3.getTextContent();

        System.out.printf(format: "User id = %s\n", uid);
        System.out.printf(format: "Firstname = %s\n", fname);
        System.out.printf(format: "Lastname = %s\n", lname);
        System.out.printf(format: "Profession = %s\n", nation);
    }
}
}
}
}

```

## 2.Feladat Adatlekérdezés

```

package hu.dompars.parse.wh85zh;

import java.io.File;
import java.io.IOException;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;

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

public class DOMQueryWH85ZH {

    Run | Debug
    public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException {
        //Forrás file
        File file = new File(pathname: "XMLWH85ZH.xml");

        //csapatok által szerzett összpontszám
        int sum = 0;

        DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();

        Document doc = dBuilder.parse(file);
        doc.getDocumentElement().normalize();
        //Gyökér elem
        System.out.print(s: "Root element: ");
        System.out.println(doc.getDocumentElement().getNodeName());
        //Tabella mentése
        NodeList nList = doc.getElementsByTagName(tagname: "tabella");

```

```

        System.out.println(x: "-----");

        //Végigfut a tabella gyerek elemein kihagyva a pontokat
        for (int i = 0; i < nList.getLength(); i++) {
            Node node = nList.item(i);
            System.out.println("\nCurrent Element : "+node.getNodeName());
            if (node.getNodeType() == Node.ELEMENT_NODE) {
                Element elem = (Element) node;
                System.out.println("ID: "+elem.getAttribute(name: "T_ID"));
                NodeList nList2 = elem.getChildNodes();
                for (int j = 0; j < nList2.getLength(); j++) {
                    Node node2 = nList2.item(j);
                    if (node2.getNodeType() == Node.ELEMENT_NODE) {
                        Element elem2 = (Element) node2;
                        if(!node2.getNodeName().equals(anObject: "pontok")) {
                            System.out.println(node2.getNodeName()+" : "+node2.getTextContent());
                        }else{
                            sum += Integer.parseInt(node2.getTextContent());
                        }
                        NodeList nList3 = elem2.getChildNodes();
                        for (int k = 0; k < nList3.getLength(); k++) {
                            Node node3 = nList3.item(k);
                            if(node3.getNodeType()==Node.ELEMENT_NODE) {
                                System.out.println("pontok : " +node3.getNodeName()+" : "+node3.getTextContent());
                            }
                        }
                    }
                }
            }
        }
        //összpontok
        System.out.println("Összes pont: "+ sum);
    }
}

```

## 2.Feladat Adatmódosítás

```
package hu.domparse.wh85zh;

import java.io.File;
import java.io.IOException;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;

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

public class DOMModifyWH85ZH {

    Run | Debug
    public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException {

        try {
            //Forrás file
            File inputFile = new File(pathname: "XML2WH85ZH.xml");
            DocumentBuilderFactory docFactory = DocumentBuilderFactory.newInstance();
            DocumentBuilder docBuilder = docFactory.newDocumentBuilder();
            Document doc = docBuilder.parse(inputFile);

            //Második és harmadik egyesület mentése
            Node club1 = doc.getElementsByTagName(tagname: "egyesulet").item(index: 1);
            Node club2 = doc.getElementsByTagName(tagname: "egyesulet").item(index: 2);
            //Gyökérelem
            Node bajnoksag = doc.getFirstChild();
```



```

//Harmadik egyesület ID váltás
NamedNodeMap attr = club2.getAttributes();
Node nodeAttr = attr.getNamedItem(name: "E_ID");
nodeAttr.setTextContent(textContent: "4");

//Második egyesület irányítószámának megváltoztatása
NodeList list = club1.getChildNodes();
for (int i = 0; i < list.getLength(); i++) {
    Node node = list.item(i);
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element elem = (Element) node;
        if ("iranyitoszam".equals(elem.getNodeName())) {
            elem.setTextContent(textContent: "3532");
        }
    }
}

//Harmadik egyesület utcanevének megváltoztatása
NodeList list1 = club2.getChildNodes();
for (int i = 0; i < list1.getLength(); i++) {
    Node node1 = list1.item(i);
    if (node1.getNodeType() == Node.ELEMENT_NODE) {
        Element elem1 = (Element) node1;
        if ("utca".equals(elem1.getNodeName())) {
            elem1.setTextContent(textContent: "Gorton");
        }
    }
}

//Bírók kitörlése
NodeList childNodes = bajnoksag.getChildNodes();
for(int i = 0; i < childNodes.getLength(); i++) {
    Node node = childNodes.item(i);

    if("biro".equals(node.getNodeName()))
        bajnoksag.removeChild(node);
}

```