

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Könyvtári nyilvántartás létrehozása

Készítette: **Vékony Róbert**

Neptunkód: **H0F0SZ**

Dátum: **2023.12.14**

Tartalom

1. Feladat.....	4
1.1 A feladat leírása:.....	4
1.2 Az ER modell egyedei és tulajdonságai:	4
1.3 Az XDM modell alapján XML dokumentum készítése:.....	7
1.3.1. Statisztikák.....	7
1.3.2. Labdarúgók.....	8
1.3.3. Csapatok.....	8
1.3.4. Mérkőzések	9
1.3.6. Mérkőzésen játszó csapatok	10
1.4. Az XML dokumentum alapján XML Schema készítése:	11
1.4.1. Kor saját típus létrehozása	11
1.4.2. Poszt saját típus létrehozása	11
1.4.3. Pont saját típus létrehozása	11
1.4.4. Bajnokság felépítése (complex típus).....	12
1.4.5. Kulcsok létrehozása (referencia)	14
1.4.6. Idegen kulcsok létrehozása	15
1.4.7. Unique kapcsolat létrehozása	15
2. A DOM program készítése.....	16
2.1. Adatolvasás DomReadHOFOSZ.....	16
2.1.1. File megnyitása, és feldolgozása.	16
2.1.2. Elemeken végigiterálás és kiírása.	17
2.1.3. Az eredmény.....	18
2.2. Adatmódosítás DomModifyHOFOSZ	21
2.2.1. XML file betöltése.....	21
2.2.2. Statisztika node-ok módosítása, játékerp beállítása 300-ra.....	21
2.2.3. Stadion node-ok módosítása.....	21
2.2.4. Mérkőzés módosítása.....	22
2.2.5. A frissen módosított XML file tartalmának mentése és kiírása konzolra.....	22
2.2.6. Eremény (részlet)	22
2.3. Adatlekérdezés DomQueryHOFOSZ	23
2.3.1. XML file betöltése.....	23
2.3.2. labdarugók neveinek lekérdezése és kiírása	23
2.3.3. "I2" ID-jú Labdarugó játszott perceinek a lekérdezése és kiírása	23

2.3.3. Csapatok lekérdezése és kiírása	24
2.3.4. Stadionok neveinek és férőhelyeinek lekérdezése és kiírása	24
2.3.5. Mérkőzések idelyének lekérdezése és kiírása.....	24
2.3.6. Eredmény	25
2.4. Adatírás DomWriteHOF0SZ.....	26
2.4.1. XML dokumentum inicializálása és root element létrehozása.....	26
2.4.3. A Segédfüggvényeket használva az elemek feltöltése	27
2.4.4. Transzformáció, majd konzolra és fileba írás	27

1. Feladat

1.1 A feladat leírása:

A feladat egy hiteles adatbázis létrehozása, különböző lekérdezések kialakítása XML nyelven keresztül. A beadandó feladat témája a profi angol bajnokságokban szereplő csapatok és játékosok, valamint meccseiknek tárolására alkalmas adatbázis. Továbbá tartalmaz a mérkőzések helyszínéül szolgáló stadionokról információkat, valamint részletes játékos statisztikákat. Ezután az adatbázis szerkezetét leíró ER, valamint XDM modelljének kialakítása a következő lépés. Amint a megfelelő konvertálások megtörténnek a feladat folytatódik az xml fájl létrehozásával és feltöltésével. Az XML dokumentum feltöltése után egy XML Schema kell létre hozni.

1.2 Az ER modell egyedei és tulajdonságai:

Egyedek: Csapat, Stadion, Merkozes, Labdarugo, Statisztika

Egyedek tulajdonságai:

Csapat:

- CsapatID: A csapat egyed elsődleges kulcsa
- Edzo: A csapat edzője
- CsNev: A csapat neve
- Liga: A bajnokság neve, ahol a csapat játszik, több értékű tulajdonság.

Stadion:

- StadionID: A stadion egyed elsődleges kulcsa
- Nev: A stadion neve
- Cim: A stadion pontos címe
- Ferohely: A maximális nézőszám a stadionban

Merkozes:

- MerkozesID: A mérkőzés egyed elsődleges kulcsa

- HazaiGolk: A hazai csapat által szerzett gólok mennyisége
- VendegGolk: A vendégcsapat által szerzett gólok mennyisége
- Idopont: A mérkőzés időpontja

Labdarugo:

- LabdarugoID: A labdarúgó egyed elsődleges kulcsa
- LNev: A játékos neve
- Kor: A labdarúgó kora
- Poszt: A labdarúgó posztja

Statisztika:

- StatisztikaID: A statisztika egyed elsődleges kulcsa
- JatekPerc: A labdarúgó pályán töltött perceinek mennyisége
- SargaLap: A labdarúgó sárga lapjainak mennyisége
- PirosLap: A labdarúgó piros lapjainak mennyisége
- Gol: Góljainak száma
- Golpassz: Gólpaszainak száma

Egyedek közötti kapcsolat:

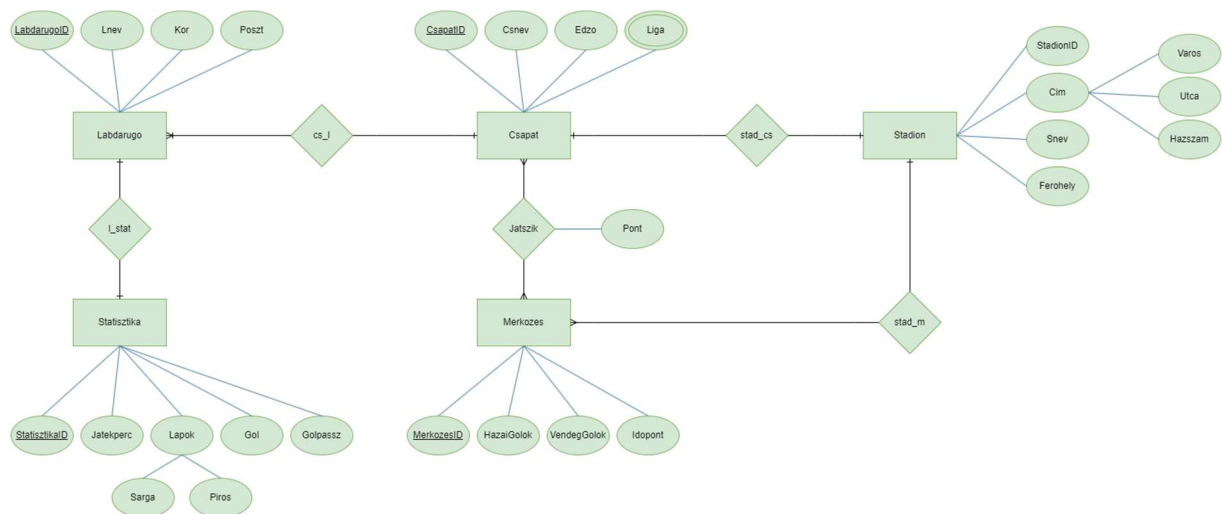
A **Labdarúgó** és a **Statisztika** egyedek között 1:1 kapcsolat van, ugyanis minden játékos rendelkezik a saját statisztikájával, a statisztikához pedig mindig egy adott játékos tartozik.

A **Labdarúgó** és a **Csapat** egyedek között 1:N kapcsolat van, ugyanis a csapatok több játékosal rendelkeznek, de minden játékos egy csapathoz van leszerződve.

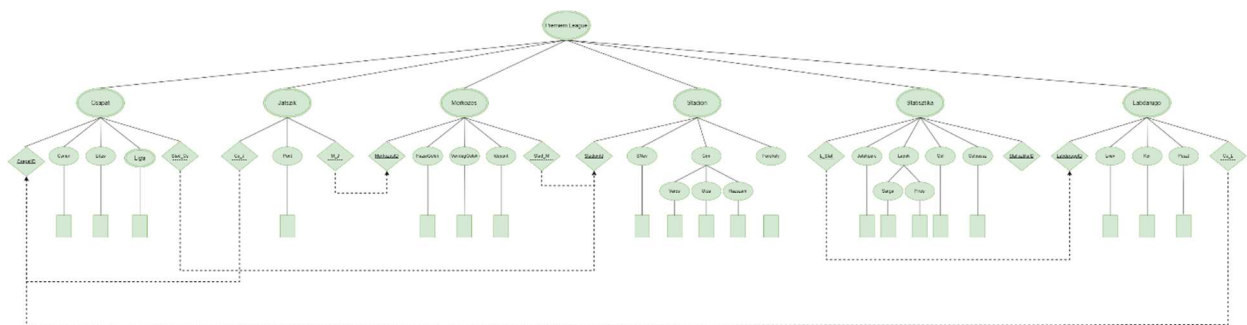
A **Csapat** és a **Stadion** egyedek között 1:1 kapcsolat van, mert minden csapat rendelkezik egy stadionnal, és minden stadionnak is pontosan egy csapata van, akinek ez a hazai pályája.

A **Stadion** és a **Mérkőzés** egyedek között 1:N kapcsolat van, mert egy mérkőzés nyilvánvalóan egy bizonyos stadionban játszódik, azonban egy stadionban számos mérkőzés lejátszható.

A **Csapat** és a **Mérkőzés** egyedek között N:M kapcsolat van, mert egy mérkőzést több csapat játszik és egy csapat értelemszerűen sok mérkőzést játszik egy szezon alatt is.



Az XDM modellben a háromfajta jelölést használunk. Az ER modellben szereplő elemeket az XDM modellben ellipszissel, az elemek tulajdonságait, azaz az attribútumokat rombusszal, a szöveges tartalmakat pedig téglalappal jelöljük. A kulcstulajdonságok az XDM modellben is aláhúzásra kerülnek, valamint külön elemet kell létrehozni a több-több kapcsolat szemléltetésére.



1.3 Az XDM modell alapján XML dokumentum készítése:

Az XDM modell alapján létrehoztam az XML dokumentumot. Minden egyedhez előállítottam legalább három példát. Nagyjából helyes de nem a valóságot tükröző adatokat tartalmaz a dokumentum.

1.3.1. Statisztikák

```
<!-- Statisztikák -->
<statisztika statisztikaID="stat1" l_stat="l1">
  <jatekperc>300</jatekperc>
  <lapok>
    <sarga>4</sarga>
    <piros>0</piros>
  </lapok>
  <gol>4</gol>
  <golpassz>5</golpassz>
</statisztika>

<statisztika statisztikaID="stat2" l_stat="l2">
  <jatekperc>1000</jatekperc>
  <lapok>
    <sarga>2</sarga>
    <piros>0</piros>
  </lapok>
  <gol>10</gol>
  <golpassz>13</golpassz>
</statisztika>

<statisztika statisztikaID="stat3" l_stat="l3">
  <jatekperc>600</jatekperc>
  <lapok>
    <sarga>10</sarga>
    <piros>1</piros>
  </lapok>
  <gol>1</gol>
  <golpassz>0</golpassz>
</statisztika>
```

1.3.2. Labdarúgók

```
<!-- Labdarúgók -->
  <labdarugo labdarugoID="l1" cs_l="cs1">
    <lnev>James Maddison</lnev>
    <kor>22</kor>
    <poszt>CAM</poszt>
  </labdarugo>

  <labdarugo labdarugoID="l2" cs_l="cs2">
    <lnev>Gabriel Jesus</lnev>
    <kor>24</kor>
    <poszt>ST</poszt>
  </labdarugo>

  <labdarugo labdarugoID="l3" cs_l="cs3">
    <lnev>Ruben Dias</lnev>
    <kor>26</kor>
    <poszt>CB</poszt>
  </labdarugo>
```

1.3.3. Csapatok

```
<!-- Csapatok -->
  <csapat csapatID="cs1" stad_cs="s1">
    <csnev>Leicester City</csnev>
    <edzo>Jose Mourinho</edzo>
    <liga>Premier Leauge</liga>
    <liga>Europe Leauge</liga>
  </csapat>

  <csapat csapatID="cs2" stad_cs="s2">
    <csnev>Arsenal</csnev>
    <edzo>Mikel Arteta</edzo>
    <liga>Premier Leauge</liga>
    <liga>Europe Leauge</liga>
  </csapat>

  <csapat csapatID="cs3" stad_cs="s3">
    <csnev>Manchester City</csnev>
    <edzo>Pep Guardiola</edzo>
    <liga>Premier Leauge</liga>
    <liga>Champions Leauge</liga>
  </csapat>
```


1.3.4. Mérkőzések

```
<!-- Mérkőzések -->
  <merkozes merkozesID="m1" stad_m="s1">
    <hazaigolok>2</hazaigolok>
    <vendeggolok>0</vendeggolok>
    <idopont>2022/10/20 18:00</idopont>
  </merkozes>

  <merkozes merkozesID="m2" stad_m="s2">
    <hazaigolok>0</hazaigolok>
    <vendeggolok>0</vendeggolok>
    <idopont>2022/09/12 20:00</idopont>
  </merkozes>

  <merkozes merkozesID="m3" stad_m="s2">
    <hazaigolok>1</hazaigolok>
    <vendeggolok>2</vendeggolok>
    <idopont>2022/09/30 21:00</idopont>
  </merkozes>
```

1.3.5. Stadionok

```
<!-- Stadionok -->
  <stadion stadionID="s1"><snev>King Power</snev>
    <cim>
      <varos>Leicester</varos>
      <utca>Eper utca</utca>
      <hazszam>12</hazszam>
    </cim>
    <ferohely>40000</ferohely>
  </stadion>

  <stadion stadionID="s2">
    <snev>Emirates Stadium</snev>
    <cim>
      <varos>London</varos>
      <utca>Kakas utca</utca>
      <hazszam>30</hazszam>
    </cim>
    <ferohely>80000</ferohely>
  </stadion>

  <stadion stadionID="s3">
    <snev>Etihad</snev>
```

1.3.6. Mérkőzésen játszó csapatok

```
<!-- Mérkőzésen játszó csapatok -->
<jatszik cs_j="cs1" m_j="m1">
  <pont>0</pont>
</jatszik>

<jatszik cs_j="cs2" m_j="m1">
  <pont>3</pont>
</jatszik>

<jatszik cs_j="cs3" m_j="m2">
  <pont>1</pont>
</jatszik>

<jatszik cs_j="cs1" m_j="m2">
  <pont>1</pont>
</jatszik>

<jatszik cs_j="cs2" m_j="m3">
  <pont>0</pont>
</jatszik>

<jatszik cs_j="cs3" m_j="m3">
  <pont>3</pont>
</jatszik>
```

1.4. Az XML dokumentum alapján XML Schema készítése:

Első lépés a fejléc megadása volt.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
```

1.4.1. Kor saját típus létrehozása

```
<!-- Saját típusok -->
<xs:element name="kor">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="100"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

1.4.2. Poszt saját típus létrehozása

```
<xs:simpleType name="poszt">
  <xs:restriction base="xs:string">
    <xs:enumeration value="CAM" />
    <xs:enumeration value="CB" />
    <xs:enumeration value="ST" />
    <xs:enumeration value="CDM" />
    <xs:enumeration value="GK" />
    <xs:enumeration value="CM" />
    <xs:enumeration value="RM" />
    <xs:enumeration value="LM" />
    <xs:enumeration value="RW" />
    <xs:enumeration value="LW" />
  </xs:restriction>
</xs:simpleType>
```

1.4.3. Pont saját típus létrehozása

```
<xs:simpleType name="pont">
  <xs:restriction base="xs:integer">
    <xs:enumeration value="0" />
    <xs:enumeration value="1" />
    <xs:enumeration value="3" />
  </xs:restriction>
</xs:simpleType>
```

1.4.4. Bajnokság felépítése (complex típus)

```
<!-- Felépítés -->
<xs:element name="bajnoksag">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="statisztika">
        <xs:complexType>
          <xs:sequence>
            <xs:element type="xs:integer" name="jatekperc" />
            <xs:element name="lapok">
              <xs:complexType>
                <xs:sequence>
                  <xs:element type="xs:integer"
name="sarga"/>
                  <xs:element type="xs:integer"
name="piros"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element type="xs:integer" name="gol" />
            <xs:element type="xs:integer" name="golpassz" />
          </xs:sequence>
          <xs:attribute type="xs:string" name="l_stat"
use="required" />
          <xs:attribute type="xs:string" name="statisztikaID"
use="required" />
        </xs:complexType>
      </xs:element>
      <xs:element name="labdarugo">
        <xs:complexType>
          <xs:sequence>
            <xs:element type="xs:string" name="lnev" />
            <xs:element type="kor" name="kor" />
            <xs:element type="poszt" name="poszt" />
          </xs:sequence>
          <xs:attribute type="xs:string" name="cs_l"
use="required" />
          <xs:attribute type="xs:string" name="labdarugoID"
use="required" />
        </xs:complexType>
      </xs:element>
      <xs:element name="csapat">
        <xs:complexType>
          <xs:sequence>
            <xs:element type="xs:string" name="csnev" />
            <xs:element type="xs:string" name="edzo" />
            <xs:element type="xs:string" name="liga" />
          </xs:sequence>
          <xs:attribute type="xs:string" name="stad_cs"
use="required" />
          <xs:attribute type="xs:string" name="csapatID"
use="required" />
        </xs:complexType>
      </xs:element>
      <xs:element name="merkozes">
        <xs:complexType>
```

```

        <xs:sequence>
            <xs:element type="xs:integer" name="hazaigolok" />
            <xs:element type="xs:integer" name="vendeggolok"
/>
            <xs:element type="xs:date" name="idopont" />
        </xs:sequence>
        <xs:attribute type="xs:string" name="stad_m"
use="required" />
        <xs:attribute type="xs:string" name="merkozesID"
use="required" />
    </xs:complexType>
</xs:element>
<xs:element name="stadion">
    <xs:complexType>
        <xs:sequence>
            <xs:element type="xs:string" name="snev" />
            <xs:element name="cim">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element type="xs:string"
name="varos"/>
                        <xs:element type="xs:string"
name="utca"/>
                        <xs:element type="xs:integer"
name="hazszam"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element type="xs:integer" name="ferohely" />
        </xs:sequence>
        <xs:attribute type="xs:string" name="stadionID"
use="required" />
    </xs:complexType>
</xs:element>
<xs:element name="jatszik">
    <xs:complexType>
        <xs:sequence>
            <xs:element type="pont" name="pont" />
        </xs:sequence>
        <xs:attribute type="xs:string" name="cs_j"
use="required" />
        <xs:attribute type="xs:string" name="m_j"
use="required" />
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>

```

1.4.5. Kulcsok létrehozása (referencia)

```
<!-- Kulcsok -->
  <xs:key name = "statisztika_kulcs">
    <xs:selector xpath="statisztika"/>
    <xs:field xpath = "@statisztikaID" />
  </xs:key>

  <xs:key name = "labdarugo_kulcs">
    <xs:selector xpath="labdarugo"/>
    <xs:field xpath = "@labdarugoID" />
  </xs:key>

  <xs:key name = "csapat_kulcs">
    <xs:selector xpath="csapat"/>
    <xs:field xpath = "@csapatID" />
  </xs:key>

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

  <xs:key name = "stadion_kulcs">
    <xs:selector xpath="stadion"/>
    <xs:field xpath = "@stadionID" />
  </xs:key>
```

1.4.6. Idegen kulcsok létrehozása

```
<!-- Idegen kulcsok -->
<xs:keyref refer = "labdarugo_kulcs" name = "labdarugo_idegen_kulcs">
  <xs:selector xpath = "statisztika" />
  <xs:field xpath = "@l_stat" />
</xs:keyref>

<xs:keyref refer = "csapat_kulcs" name = "csapat_idegen_kulcs">
  <xs:selector xpath = "labdarugo" />
  <xs:field xpath = "@cs_l" />
</xs:keyref>

<xs:keyref refer = "stadion_kulcs" name
="stadion_csapat_idegen_kulcs">
  <xs:selector xpath = "csapat" />
  <xs:field xpath = "@stad_cs" />
</xs:keyref>

<xs:keyref refer = "stadion_kulcs" name
="stadion_csapat_idegen_kulcs">
  <xs:selector xpath = "merkozes" />
  <xs:field xpath = "@stad_m" />
</xs:keyref>

<xs:keyref refer = "csapat_kulcs" name = "csapat_jatsziz_idegen_kulcs">
  <xs:selector xpath = "jatsziz" />
  <xs:field xpath = "@cs_j" />
</xs:keyref>

<xs:keyref refer = "merkozes_kulcs" name
="merkozes_jatsziz_idegen_kulcs">
  <xs:selector xpath = "jatsziz" />
  <xs:field xpath = "@m_j" />
</xs:keyref>
</xs:element>
```

1.4.7. Unique kapcsolat létrehozása

```
<xs:unique name="unique_stadion_csapat">
  <xs:selector xpath="csapat" />
  <xs:field xpath="@stad_cs" />
</xs:unique>
```

2. A DOM program készítése.

A feladat egy DOM program elkészítése volt az elkészített XMLH0F0SZ.xml adatainak adminisztrálására.

2.1. Adatolvasás DomReadH0F0SZ

2.1.1. File megnyitása, és feldolgozása.

```
public class DomReadH0F0SZ {  
    public static void main(String[] args) throws SAXException,  
        IOException, ParserConfigurationException, TransformerException  
    {  
        //File megnyitás  
        File xmlFile = new File("XMLH0F0SZ.xml");  
  
        //Builderek létrehozása  
        DocumentBuilderFactory factory =  
DocumentBuilderFactory.newInstance();  
        DocumentBuilder dBuilder = factory.newDocumentBuilder();  
  
        //Új és meglévő dokumentum kiválasztás  
        Document doc = dBuilder.parse(xmlFile);  
  
        doc.getDocumentElement().normalize();  
  
        //Transzformáció  
        TransformerFactory transformerFactory =  
TransformerFactory.newInstance();  
        Transformer transf = transformerFactory.newTransformer();  
  
        transf.setOutputProperty(OutputKeys.ENCODING, "UTF-8");  
        transf.setOutputProperty(OutputKeys.INDENT, "yes");  
        transf.setOutputProperty("{https://xml.apache.org/xslt}indent-  
amount", "2");  
  
        //Gyökér elem kiírás  
        System.out.println("Root element: " +  
doc.getDocumentElement().getNodeName());  
    }  
}
```


2.1.2. Elemeken végigiterálás és kiírása.

```
//Gyökér elem kiírás
    System.out.println("Root element: " +
doc.getDocumentElement().getNodeName());

//Adatok kigyűjtése az XML-ből
NodeList statList = doc.getElementsByTagName("statisztika");
for(int i = 0; i < statList.getLength(); i++) {
    Node nNode = statList.item(i);

    System.out.println("\nCurrent Element: " + nNode.getNodeName());

    if( nNode.getNodeType() == Node.ELEMENT_NODE) {
        Element elem = (Element) nNode;

        String sid = elem.getAttribute("statisztikaID");
        String lstatid = elem.getAttribute("l_stat");

        Node node1 = elem.getElementsByTagName("jatekperc").item(0);
        String mins = node1.getTextContent();

        Node node2 = elem.getElementsByTagName("sarga").item(0);
        Node node3 = elem.getElementsByTagName("piros").item(0);
        String yellow = node2.getTextContent();
        String red = node3.getTextContent();

        Node node4 = elem.getElementsByTagName("gol").item(0);
        String goals = node4.getTextContent();

        Node node5 = elem.getElementsByTagName("golpassz").item(0);
        String assists = node5.getTextContent();

        System.out.println("Statistics id: " + sid);
        System.out.println("Played minutes: " + mins);
        System.out.println("Yellow cards: " + yellow);
        System.out.println("Red cards: " + red);
        System.out.println("Goals: " + goals);
        System.out.println("Assists: " + assists);
        System.out.println("Foreign key to football player: " +
lstatid);
    }
}
```

2.1.3. Az eredmény

Root element: bajnoksag

Current Element: statisztika

Statistics id: stat1

Played minutes: 300

Yellow cards: 4

Red cards: 0

Goals: 4

Assits: 5

Foreign key to football player: 11

Current Element: statisztika

Statistics id: stat2

Played minutes: 1000

Yellow cards: 2

Red cards: 0

Goals: 10

Assits: 13

Foreign key to football player: 12

Current Element: statisztika

Statistics id: stat3

Played minutes: 600

Yellow cards: 10

Red cards: 1

Goals: 1

Assits: 0

Foreign key to football player: 13

Current Element: labdarugo

Football player id: 11

Name: James Maddison

Age: 22

Position: CAM

Foreign key to team: cs1

Current Element: labdarugo

Football player id: 12

Name: Gabriel Jesus

Age: 24

Position: ST

Foreign key to team: cs2

Current Element: labdarugo

Football player id: 13

Name: Ruben Dias

Age: 26

Position: CB

Foreign key to team: cs3

Current Element: csapat

Team id: cs1

Name: Leicester City
Manager: Jose Mourinho
League: Premier League
League: Europe League
Foreign key to stadium: s1

Current Element: csapat
Team id: cs2
Name: Arsenal
Manager: Mikel Arteta
League: Premier League
League: Europe League
Foreign key to stadium: s2

Current Element: csapat
Team id: cs3
Name: Manchester City
Manager: Pep Guardiola
League: Premier League
League: Champions League
Foreign key to stadium: s3

Current Element: merkozés
Match id: m1
Home goals: 2
Away goals: 0
Start time: 2022/10/20 18:00
Foreign key to stadium: s1

Current Element: merkozés
Match id: m2
Home goals: 0
Away goals: 0
Start time: 2022/09/12 20:00
Foreign key to stadium: s2

Current Element: merkozés
Match id: m3
Home goals: 1
Away goals: 2
Start time: 2022/09/30 21:00
Foreign key to stadium: s2

Current Element: stadion
Game id: s1
Name: King Power
City: Leicester
Street: Eper utca
House Number: 12
Capacity: 40000

Current Element: stadion
Game id: s2

Name: Emirates Stadium
City: London
Street: Kakas utca
House Number: 30
Capacity: 80000

Current Element: stadion
Game id: s3
Name: Etihad
City: Manchester
Street: Kossuth utca
House Number: 16
Capacity: 60000

Current Element: jatszík
Team id: cs1
Game id: m1
Earned point(s): 0

Current Element: jatszík
Team id: cs2
Game id: m1
Earned point(s): 3

Current Element: jatszík
Team id: cs3
Game id: m2
Earned point(s): 1

Current Element: jatszík
Team id: cs1
Game id: m2
Earned point(s): 1

Current Element: jatszík
Team id: cs2
Game id: m3
Earned point(s): 0

Current Element: jatszík
Team id: cs3
Game id: m3
Earned point(s): 3

2.2. Adatmódosítás DomModifyH0F0SZ

2.2.1. XML file betöltése

```
File inputFile = new File("XMLH0F0SZ.xml");
DocumentBuilderFactory documentBuilderFactory =
DocumentBuilderFactory.newInstance();

DocumentBuilder documentBuilder =
documentBuilderFactory.newDocumentBuilder();

Document doc = documentBuilder.parse(inputFile);
```

2.2.2. Statisztika node-ok módosítása, játékperc beállítása 300-ra.

```
// statisztika módosítása

Node stat = doc.getElementsByTagName("statisztika").item(0);
NodeList statList = stat.getChildNodes();

for (int i = 0; i < statList.getLength(); i++) {
    Node node = statList.item(i);
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element eElement = (Element) node;

        if ("jatekperc".equals(eElement.getNodeName())) {
            eElement.setTextContent("300");
        }
    }
}
```

2.2.3. Stadion node-ok módosítása

Ha 65000 férőhelyes akkor 60000-re. Ha 39000 akkor 40000-re.

```
//stadion módosítása

Node stad = doc.getElementsByTagName("stadion").item(0);
NodeList stadlist = stad.getChildNodes();

for (int i = 0; i < stadlist.getLength(); i++) {
    Node node = stadlist.item(i);
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element eElement = (Element) node;
        if ("ferohely".equals(eElement.getNodeName())) {
            if ("65000".equals(eElement.getTextContent())) {
                eElement.setTextContent("60000");
            }
            if ("39000".equals(eElement.getTextContent())) {
                eElement.setTextContent("40000");
            }
        }
    }
}
```

2.2.4. Mérkőzés módosítása

Hazai gólok beállítása 3-ról 2-re és vendéggólok 1-ről 0-ra.

```
//merkozes módosítása
Node merk = doc.getElementsByTagName("merkozes").item(0);
NodeList merklist = merk.getChildNodes();

for (int i = 0; i < merklist.getLength(); i++) {
    Node node = merklist.item(i);
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element eElement = (Element) node;
        if ("hazaigolok".equals(eElement.getNodeName())) {
            if ("3".equals(eElement.getTextContent())) {
                eElement.setTextContent("2");
            }
        }
        if ("vendeggolok".equals(eElement.getNodeName())) {
            if ("1".equals(eElement.getTextContent())) {
                eElement.setTextContent("0");
            }
        }
    }
}
```

2.2.5. A frissen módosított XML file tartalmának mentése és kiírása konzolra.

```
// Tartalom konzolra és fájlba való írása

TransformerFactory transformerFactory =
TransformerFactory.newInstance();
Transformer transformer = transformerFactory.newTransformer();

DOMSource source = new DOMSource(doc);

System.out.println("----Módosított fájl----");

StreamResult consoleResult = new StreamResult(System.out);
StreamResult file = new StreamResult(inputFile);

transformer.transform(source, consoleResult);
transformer.transform(source, file);
```

2.2.6. Eremény (részlet)

```
----Módosított fájl----
<?xml version="1.0" encoding="UTF-8" standalone="no"?><bajnoksag>
  <!-- Statisztikák -->
  <statisztika l_stat="11" statisztikaID="stat1">
    <jatekperc>300</jatekperc>
    <lapok>
      <sarga>4</sarga>
      <piros>0</piros>
    </lapok>
    <gol>4</gol>
    <golpassz>5</golpassz>
  </statisztika>
```

2.3. Adatlekérdezés DomQueryH0F0SZ

Az XML dokumentum példányai közül 5 lekérdezése és kiírása konzolra.

2.3.1. XML file betöltése

```
DocumentBuilderFactory documentBuilderFactory =
DocumentBuilderFactory.newInstance();
DocumentBuilder documentBuilder =
documentBuilderFactory.newDocumentBuilder();

Document document =documentBuilder.parse("XMLH0F0SZ.xml");

document.getDocumentElement().normalize();
```

2.3.2. labdarugók neveinek lekérdezése és kiírása

```
// 1) Labdarugok neveinek lekerdezese
NodeList labdarugoList =
document.getElementsByTagName("labdarugo");
System.out.println("Labdarugok nevei:");
for (int i = 0; i < labdarugoList.getLength(); i++) {
    Node node = labdarugoList.item(i);
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element labdarugo = (Element) node;

System.out.println(labdarugo.getElementsByTagName("lnev").item(0).getTextCo
ntent());
    }
}
```

2.3.3. "12" ID-jú Labdarugó játszott perceinek a lekérdezése és kiírása

```
// 2) 12 ID-ju Labdarugo játszott perceinek a lekerdezese
String labdarugoIdToQuery = "12";
NodeList statisztikaList =
document.getElementsByTagName("statisztika");
for (int i = 0; i < statisztikaList.getLength(); i++) {
    Node node = statisztikaList.item(i);
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element statisztika = (Element) node;
        if
(statisztika.getAttribute("l_stat").equals(labdarugoIdToQuery)) {
            System.out.println("Az 12 ID-ju labdarugo játszott
percei: " +

            statisztika.getElementsByTagName("jatekperc").item(0).getTextContent());
            break;
        }
    }
}
```

2.3.3. Csapatok lekérdezése és kiírása

```
// 3) Csapatok lekerdezese
NodeList csapatList = document.getElementsByTagName("csapat");
System.out.println("Rendszerben szereplő csapatok:");
for (int i = 0; i < csapatList.getLength(); i++) {
    Node node = csapatList.item(i);
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element csapat = (Element) node;

        System.out.println(csapat.getElementsByTagName("csnev").item(0).getTextContent());
    }
}
```

2.3.4. Stadionok neveinek és férőhelyeinek lekérdezése és kiírása

```
// 4) Stadionok es ferohelyeik szama
NodeList stadionList =
document.getElementsByTagName("stadion");
System.out.println("Stadionok és ferohelyeik szama:");
for (int i = 0; i < stadionList.getLength(); i++) {
    Node node = stadionList.item(i);
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element stadion = (Element) node;
        System.out.println(

        stadion.getElementsByTagName("snev").item(0).getTextContent()
            +" Ferohely: "+

        stadion.getElementsByTagName("ferohely").item(0).getTextContent()
        );
    }
}
```

2.3.5. Mérkőzések idelyének lekérdezése és kiírása

```
// 5) merkozesek idelyenek lekerdezese
NodeList merkozesList =
document.getElementsByTagName("merkozes");
System.out.println("A merkozesek idopontjai:");
for (int i = 0; i < merkozesList.getLength(); i++) {
    Node node = merkozesList.item(i);
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element merkozes = (Element) node;
        System.out.println(

        merkozes.getElementsByTagName("idopont").item(0).getTextContent()
        );
    }
}
```


2.3.6. Eredmény

Labdarugók nevei:

James Maddison

Gabriel Jesus

Ruben Dias

Az 12 ID-ju labdarugo játszott percei: 1000

Rendszerben szereplő csapatok:

Leicester City

Arsenal

Manchester City

Stadionok és ferohelyeik szama:

King Power Ferohely: 40000

Emirates Stadium Ferohely: 80000

Etihad Ferohely: 60000

A merkozések idopontjai:

2022/10/20 18:00

2022/09/12 20:00

2022/09/30 21:00

2.4. Adatírás DomWriteHOFOSZ

XML Struktúra felépítése és feltöltése adatokkal a DOM programmal, majd fileba írása.

2.4.1. XML dokumentum inicializálása és root element létrehozása

```
//Builderek létrehozása
    DocumentBuilderFactory factory =
DocumentBuilderFactory.newInstance();
    DocumentBuilder dBuilder = factory.newDocumentBuilder();

    //Új dokumnetum kiválasztás
    Document newdoc = dBuilder.newDocument();

    //Gyökérelem megadás
    Element root = newdoc.createElementNS("XMLHOFOSZ", "bajnoksag");
    newdoc.appendChild(root);
```

2.4.2. Segédfüggvények létrehozása az XML file írásához (példa)

```
//Segéd függvények az XML fájl íráshoz
    private static Node createStatisztika(Document newdoc, String id,
String jatekperc, String sarga, String piros, String gol, String golpassz,
String fid) {

        Element node = newdoc.createElement("statisztika");

        node.setAttribute("statisztikaID", id);
        node.setAttribute("l_stat", fid);
        node.appendChild(createElement(newdoc, "jatekperc", jatekperc));

        Node node1 = createElement(newdoc, "sarga", sarga);
        Node node2 = createElement(newdoc, "piros", piros);

        Element cards = newdoc.createElement("lapok");
        cards.appendChild(node1);
        cards.appendChild(node2);
        node.appendChild(cards);

        node.appendChild(createElement(newdoc, "gol", gol));
        node.appendChild(createElement(newdoc, "golpassz", golpassz));

        return node;
    }

    private static Node createLabdarugo(Document newdoc, String id, String
lnev, String kor, String poszt, String fid) {

        Element node = newdoc.createElement("labdarugo");

        node.setAttribute("labdarugoID", id);
        node.setAttribute("l_stat", fid);
        node.appendChild(createElement(newdoc, "lnev", lnev));
        node.appendChild(createElement(newdoc, "kor", kor));
        node.appendChild(createElement(newdoc, "poszt", poszt));

        return node;
    }
```

2.4.3. A Segédfüggvényeket használva az elemek feltöltése

```
//Új XML elemeinek a feltöltése
root.appendChild(createStatisztika(newdoc, "stat1","300", "10",
"0", "4", "10","11"));
root.appendChild(createStatisztika(newdoc, "stat2","200", "11",
"1", "0", "2","12"));
root.appendChild(createStatisztika(newdoc, "stat3","400", "0", "0",
"0", "2","13"));
root.appendChild(createLabdarugo(newdoc, "11","Callum Styles",
"25", "CM", "cs1"));
root.appendChild(createLabdarugo(newdoc, "12","Sallai Roland",
"25", "ST", "cs2"));
root.appendChild(createLabdarugo(newdoc, "13","Szoboszlai Dominik",
"21", "CAM", "cs3"));
root.appendChild(createCsapat(newdoc, "cs1","Millwall", "Kiss
Pista", new String[]{"Premier Leauge","Europe Leauge"}, "s1"));
root.appendChild(createCsapat(newdoc, "cs2","Freiburg", "Nagy
Janos", new String[]{"Premier Leauge","Europe Leauge"}, "s2"));
root.appendChild(createCsapat(newdoc, "cs3","Leipzig", "Kerek
Elemer", new String[]{"Premier Leauge","Europe Leauge"}, "s3"));
root.appendChild(createMerkozes(newdoc, "m1","0", "2", "17:00",
"s1"));
root.appendChild(createMerkozes(newdoc, "m2","1", "1", "18:00",
"s2"));
root.appendChild(createMerkozes(newdoc, "m3","3", "2", "20:00",
"s3"));
root.appendChild(createStadion(newdoc, "s1","Emirates Stadium",
"London", "Eper", "12", "60000"));
root.appendChild(createStadion(newdoc, "s2","London Stadium",
"London", "Malna", "12", "60000"));
root.appendChild(createStadion(newdoc, "s3","Etihad Stadium",
"Manchester", "Szilva", "12", "60000"));
root.appendChild(createJatszik(newdoc, "cs1","m1", "0"));
root.appendChild(createJatszik(newdoc, "cs2","m2", "1"));
root.appendChild(createJatszik(newdoc, "cs3","m3", "3"));
```

2.4.4. Transzformáció, majd konzolra és fileba írás

```
//Transzformáció
TransformerFactory transformerFactory =
TransformerFactory.newInstance();
Transformer transf = transformerFactory.newTransformer();

transf.setOutputProperty(OutputKeys.ENCODING, "UTF-8");
transf.setOutputProperty(OutputKeys.INDENT, "yes");
transf.setOutputProperty("{https://xml.apache.org/xslt}indent-
amount", "2");

//Fájl létrehozása
DOMSource source = new DOMSource(newdoc);
File myFile = new File("XMLH0F0SZ1.xml");

//Kiiratás
StreamResult file = new StreamResult(myFile);

//transf.transform(source, console);
transf.transform(source, file);
```