

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

NBA adatbázis

Készítette: **Csiszár Levente**

Neptun kód: **U678MF**

A feladat leírása:

A beadandó feladat témája az észak-amerikai profi kosárlabdliga (NBA) csapatainak, játékosainak, meccseinek tárolására alkalmas adatbázis. Tartalmaz továbbá a mérkőzések helyszínéül szolgáló stadionokról információkat, valamint részletes játékos statisztikákat.

Az adatbázisban található egyedek és azok tulajdonságai:

Csapat:

- CsapatID – A Csapat egyed elsődleges kulcsa
- Nev – A csapat fantázianeve
- Varos – A csapat által képviselt város, székhely
- Divizio – A csapat elhelyezkedése az észak amerikai kontinensen, többértékű tulajdonság

Stadion:

- StadionID – A Stadion egyed elsődleges kulcsa
- Nev – A stadion neve
- Varos – A város, ahol a stadion található
- Cím – A stadion pontos címe
- Befogadóképesség – A maximálisan nézőszám a stadionban

Meccs:

- MeccsID – A Meccs egyed elsődleges kulcsa
- Eredmeny – A mérkőzés pontos eredménye, származtatott tulajdonság
- Datum – A mérkőzés pontos dátuma

Jatekos:

- JatekosID – A játékos egyed elsődleges kulcsa
- Nev – A játékos teljes neve
- Kor – A játékos életkora
- Poszt – A játékos posztja
- Mezzsam – A játékos mezzsáma

Statisztika:

- StatisztikaID – A Statisztika egyed elsődleges kulcsa
- Mezonykosar – A meccs közben, a hárompontos vonalon belülről szerzett pontok
- Harompontos – A három pontos vonalon kívülről dobott kosarak
- Bunteto – Szabálytalanságot követően a büntetővonalról szerzett pontok
- Golpassz – Olyan passzok száma, amelyek után a csapattárs kosarat szerzett

- Lepattano – A gyűrűről lepattanó labdák megszerzésének a száma
- JatszottPerc – A játékos összes pályán töltött perce

Egyedek közötti kapcsolatok:

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

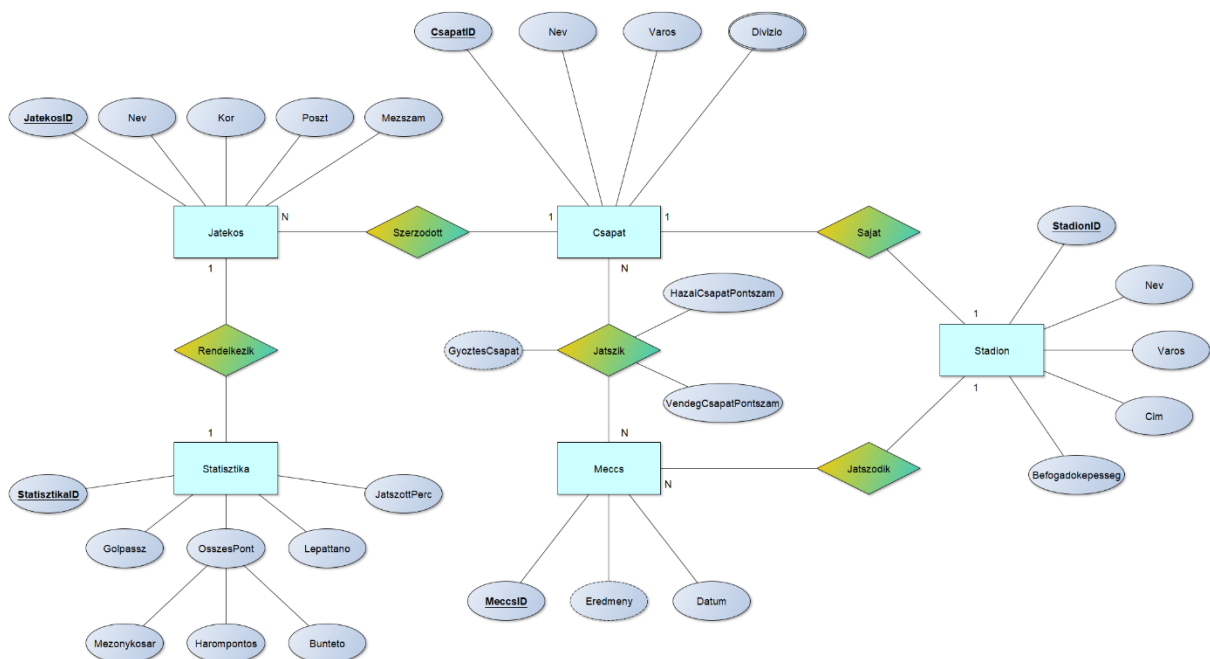
A **Csapat** és a **Stadion** egyedek között 1:1 kapcsolat van, mert minden csapat rendelkezik egy stadionnal, és minden stadionnak is pontosan egy csapata van, akinek ez a hazai pályája.

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

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

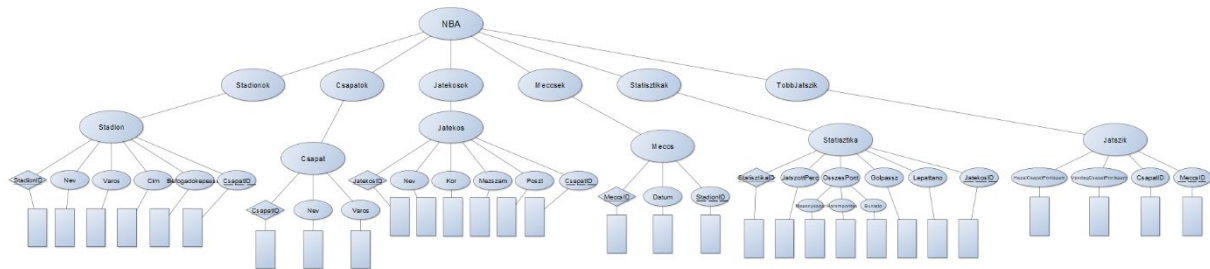
1. feladat

1a) Az adatbázis ER modell:



1b) Az adatbázis konvertálása XDM modellre:

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



1c) Az XDM modell alapján XML dokumentum készítése:

```
<?xml version="1.0" encoding="UTF-8"?>
<nba xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="XMLSchemaU678MF.xsd">
<stadionok>
  <stadion stadionID="1000">
    <nev>Staples Center</nev>
    <varos>Los Angeles</varos>
    <cim>1111 S Figueroa St</cim>
    <befogadokepesseg>20000</befogadokepesseg>
    <csapatID>2000</csapatID>
  </stadion>
  <stadion stadionID="1001">
    <nev>Fiserv Forum</nev>
    <varos>Milwaukee</varos>
    <cim>1111 Vel R. Phillips Ave</cim>
    <befogadokepesseg>17341</befogadokepesseg>
    <csapatID>2001</csapatID>
  </stadion>
  <stadion stadionID="1002">
    <nev>American Airlines Arena</nev>
    <varos>Miami</varos>
    <cim>601 Biscayne Blvd</cim>
    <befogadokepesseg>21000</befogadokepesseg>
    <csapatID>2002</csapatID>
  </stadion>
  <stadion stadionID="1003">
    <nev>TD Garden</nev>
    <varos>Boston</varos>
```

```
<cim>100 Legends Way</cim>
<befogadokepesseg>19580</befogadokepesseg>
<csapatID>2003</csapatID>
</stadion>
<stadion stadionID="1004">
  <nev>United Center</nev>
  <varos>Chicago</varos>
  <cim>1901 W Madison St</cim>
  <befogadokepesseg>23500</befogadokepesseg>
  <csapatID>2004</csapatID>
</stadion>
<stadion stadionID="1005">
  <nev>Scotiabank Arena</nev>
  <varos>Toronto</varos>
  <cim>40 Bay St</cim>
  <befogadokepesseg>19800</befogadokepesseg>
  <csapatID>2005</csapatID>
</stadion>
<stadion stadionID="1006">
  <nev>AT_T Center</nev>
  <varos>San Antonio</varos>
  <cim>1 AT_T Center Parkway</cim>
  <befogadokepesseg>18581</befogadokepesseg>
  <csapatID>2006</csapatID>
</stadion>
<stadion stadionID="1007">
  <nev>Chase Center</nev>
  <varos>San Francisco</varos>
  <cim>1 Warriors Way</cim>
  <befogadokepesseg>18064</befogadokepesseg>
  <csapatID>2007</csapatID>
</stadion>
</stadionok>
<csapatok>
  <csapat csapatID="2000">
    <nev>Lakers</nev>
    <varos>Los Angeles</varos>
  </csapat>
  <csapat csapatID="2001">
    <nev>Bucks</nev>
    <varos>Milwaukee</varos>
  </csapat>
  <csapat csapatID="2002">
    <nev>Heat</nev>
    <varos>Miami</varos>
  </csapat>
  <csapat csapatID="2003">
    <nev>Celtics</nev>
    <varos>Boston</varos>
```

```
</csapat>
<csapat csapatID="2004">
  <nev>Bulls</nev>
  <varos>Chicago</varos>
</csapat>
<csapat csapatID="2005">
  <nev>Raptors</nev>
  <varos>Toronto</varos>
</csapat>
<csapat csapatID="2006">
  <nev>Spurs</nev>
  <varos>San Antonio</varos>
</csapat>
<csapat csapatID="2007">
  <nev>Golden State Warriors</nev>
  <varos>San Francisco</varos>
</csapat>
</csapatok>
<meccsek>
  <meccs meccsID="3000">
    <datum>5/10/2021</datum>
    <stadionID>1000</stadionID>
  </meccs>
  <meccs meccsID="3001">
    <datum>5/18/2021</datum>
    <stadionID>1002</stadionID>
  </meccs>
  <meccs meccsID="3002">
    <datum>6/22/2021</datum>
    <stadionID>1003</stadionID>
  </meccs>
  <meccs meccsID="3003">
    <datum>4/12/2021</datum>
    <stadionID>1005</stadionID>
  </meccs>
  <meccs meccsID="3004">
    <datum>4/9/2021</datum>
    <stadionID>1004</stadionID>
  </meccs>
  <meccs meccsID="3005">
    <datum>5/3/2021</datum>
    <stadionID>1001</stadionID>
  </meccs>
  <meccs meccsID="3006">
    <datum>6/5/2021</datum>
    <stadionID>1005</stadionID>
  </meccs>
  <meccs meccsID="3007">
    <datum>6/2/2021</datum>
```

```
<stadionID>1006</stadionID>
</meccs>
</meccsek>
<tobbJatszok>
  <jatszok>
    <hazaiCsapatPontszam>113</hazaiCsapatPontszam>
    <vendegCsapatPontszam>96</vendegCsapatPontszam>
    <csapatID>2000</csapatID>
    <meccsID>3000</meccsID>
  </jatszok>
  <jatszok>
    <hazaiCsapatPontszam>113</hazaiCsapatPontszam>
    <vendegCsapatPontszam>96</vendegCsapatPontszam>
    <csapatID>2001</csapatID>
    <meccsID>3000</meccsID>
  </jatszok>
  <jatszok>
    <hazaiCsapatPontszam>108</hazaiCsapatPontszam>
    <vendegCsapatPontszam>106</vendegCsapatPontszam>
    <csapatID>2002</csapatID>
    <meccsID>3001</meccsID>
  </jatszok>
  <jatszok>
    <hazaiCsapatPontszam>108</hazaiCsapatPontszam>
    <vendegCsapatPontszam>106</vendegCsapatPontszam>
    <csapatID>2006</csapatID>
    <meccsID>3001</meccsID>
  </jatszok>
  <jatszok>
    <hazaiCsapatPontszam>101</hazaiCsapatPontszam>
    <vendegCsapatPontszam>101</vendegCsapatPontszam>
    <csapatID>2003</csapatID>
    <meccsID>3002</meccsID>
  </jatszok>
  <jatszok>
    <hazaiCsapatPontszam>101</hazaiCsapatPontszam>
    <vendegCsapatPontszam>101</vendegCsapatPontszam>
    <csapatID>2004</csapatID>
    <meccsID>3002</meccsID>
  </jatszok>
  <jatszok>
    <hazaiCsapatPontszam>87</hazaiCsapatPontszam>
    <vendegCsapatPontszam>110</vendegCsapatPontszam>
    <csapatID>2005</csapatID>
    <meccsID>3003</meccsID>
  </jatszok>
  <jatszok>
    <hazaiCsapatPontszam>87</hazaiCsapatPontszam>
    <vendegCsapatPontszam>110</vendegCsapatPontszam>
```

```
<csapatID>2007</csapatID>
<meccsID>3003</meccsID>
</jatszik>
<jatszik>
  <hazaiCsapatPontszam>103</hazaiCsapatPontszam>
  <vendegCsapatPontszam>105</vendegCsapatPontszam>
  <csapatID>2004</csapatID>
  <meccsID>3004</meccsID>
</jatszik>
<jatszik>
  <hazaiCsapatPontszam>103</hazaiCsapatPontszam>
  <vendegCsapatPontszam>105</vendegCsapatPontszam>
  <csapatID>2002</csapatID>
  <meccsID>3004</meccsID>
</jatszik>
</tobbJatszik>
<jatekosok>
  <jatekos jatekosID="4000">
    <nev>LeBron James</nev>
    <kor>36</kor>
    <mezzsam>23</mezzsam>
    <poszt>SF</poszt>
    <csapatID>2000</csapatID>
  </jatekos>
  <jatekos jatekosID="4001">
    <nev>Giannis Antetokounmpo</nev>
    <kor>26</kor>
    <mezzsam>34</mezzsam>
    <poszt>PF</poszt>
    <csapatID>2001</csapatID>
  </jatekos>
  <jatekos jatekosID="4002">
    <nev>Jimmy Butler</nev>
    <kor>32</kor>
    <mezzsam>22</mezzsam>
    <poszt>SG/SF</poszt>
    <csapatID>2002</csapatID>
  </jatekos>
  <jatekos jatekosID="4003">
    <nev>Jaylen Brown</nev>
    <kor>25</kor>
    <mezzsam>7</mezzsam>
    <poszt>SG/SF</poszt>
    <csapatID>2003</csapatID>
  </jatekos>
  <jatekos jatekosID="4004">
    <nev>Zach LaVine</nev>
    <kor>26</kor>
    <mezzsam>24</mezzsam>
```



```
<poszt>SG/SF</poszt>
<csapatID>2004</csapatID>
</jatekos>
<jatekos jatekosID="4005">
  <nev>Kyle Lowry</nev>
  <kor>35</kor>
  <mezzsam>7</mezzsam>
  <poszt>PG</poszt>
  <csapatID>2005</csapatID>
</jatekos>
<jatekos jatekosID="4006">
  <nev>DeMar DeRozan</nev>
  <kor>32</kor>
  <mezzsam>10</mezzsam>
  <poszt>SF/SF</poszt>
  <csapatID>2006</csapatID>
</jatekos>
<jatekos jatekosID="4007">
  <nev>DAngelo Russel</nev>
  <kor>25</kor>
  <mezzsam>0</mezzsam>
  <poszt>PG</poszt>
  <csapatID>2007</csapatID>
</jatekos>
</jatekosok>
<statisztikak>
  <statisztika statisztikaID="5000">
    <jatszottPerc>20102</jatszottPerc>
    <mezonykosar>3841</mezonykosar>
    <harompontos>2103</harompontos>
    <bunteto>3421</bunteto>
    <golpassz>1643</golpassz>
    <lepattano>1675</lepattano>
    <jatekosID>4000</jatekosID>
  </statisztika>
  <statisztika statisztikaID="5001">
    <jatszottPerc>62231</jatszottPerc>
    <mezonykosar>4232</mezonykosar>
    <harompontos>4622</harompontos>
    <bunteto>5456</bunteto>
    <golpassz>2447</golpassz>
    <lepattano>3452</lepattano>
    <jatekosID>4001</jatekosID>
  </statisztika>
  <statisztika statisztikaID="5002">
    <jatszottPerc>32131</jatszottPerc>
    <mezonykosar>1265</mezonykosar>
    <harompontos>4732</harompontos>
    <bunteto>5321</bunteto>
```

```
<golpassz>5644</golpassz>
<lepattano>3534</lepattano>
<jatekosID>4002</jatekosID>
</statisztika>
<statisztika statisztikaID="5003">
  <jatszottPerc>21333</jatszottPerc>
  <mezonykosar>4624</mezonykosar>
  <harompontos>2458</harompontos>
  <bunteto>3215</bunteto>
  <golpassz>5464</golpassz>
  <lepattano>3511</lepattano>
  <jatekosID>4003</jatekosID>
</statisztika>
<statisztika statisztikaID="5004">
  <jatszottPerc>13865</jatszottPerc>
  <mezonykosar>9754</mezonykosar>
  <harompontos>5564</harompontos>
  <bunteto>6845</bunteto>
  <golpassz>6454</golpassz>
  <lepattano>2135</lepattano>
  <jatekosID>4004</jatekosID>
</statisztika>
<statisztika statisztikaID="5005">
  <jatszottPerc>78695</jatszottPerc>
  <mezonykosar>1321</mezonykosar>
  <harompontos>5468</harompontos>
  <bunteto>5448</bunteto>
  <golpassz>5464</golpassz>
  <lepattano>3543</lepattano>
  <jatekosID>4005</jatekosID>
</statisztika>
<statisztika statisztikaID="5006">
  <jatszottPerc>21515</jatszottPerc>
  <mezonykosar>4648</mezonykosar>
  <harompontos>3213</harompontos>
  <bunteto>2135</bunteto>
  <golpassz>4421</golpassz>
  <lepattano>5443</lepattano>
  <jatekosID>4006</jatekosID>
</statisztika>
<statisztika statisztikaID="5007">
  <jatszottPerc>94656</jatszottPerc>
  <mezonykosar>6521</mezonykosar>
  <harompontos>5462</harompontos>
  <bunteto>5468</bunteto>
  <golpassz>1353</golpassz>
  <lepattano>1235</lepattano>
  <jatekosID>4007</jatekosID>
</statisztika>
```

```
</statisztikak>
</nba>
```

1d) Az XML dokumentum alapján XMLSchema készítése (saját típusok):

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="nba">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="stadionok">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="stadion" type="stadionTipus"
maxOccurs="unbounded" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>

        <xs:element name="csapatok">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="csapat" type="csapatTipus"
maxOccurs="unbounded" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>

        <xs:element name="meccsek">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="meccs" type="meccsTipus"
maxOccurs="unbounded" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>

        <xs:element name="tobbJatszik">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="jatszik" type="tobbJatszikTipus"
maxOccurs="unbounded" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>

        <xs:element name="jatekosok">
```

```

        <xs:complexType>
            <xs:sequence>
                <xs:element name="jatekos" type="jatekosTipus"
maxOccurs="unbounded" minOccurs="0" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <xs:element name="statisztikak">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="statisztika"
type="statisztikaTipus" maxOccurs="unbounded" minOccurs="0" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:sequence>
</xs:complexType>

    <xs:key name="stadionKulcs">
<xs:selector xpath="stadionok/stadion"/>
        <xs:field xpath="@stadionID"/>
    </xs:key>

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

    <xs:key name="jatekosKulcs">
        <xs:selector xpath="jatekosok/jatekos"/>
        <xs:field xpath="@jatekosID"/>
    </xs:key>

    <xs:key name="meccsKulcs">
        <xs:selector xpath="meccsek/meccs"/>
        <xs:field xpath="@meccsID"/>
    </xs:key>

    <xs:keyref name="csapat-stadion" refer="csapatKulcs">
        <xs:selector xpath="stadionok/stadion/csapatID"/>
        <xs:field xpath="."/>
    </xs:keyref>

    <xs:keyref name="stadion-meccs" refer="stadionKulcs">
        <xs:selector xpath="meccsek/meccs/stadionID"/>
        <xs:field xpath="."/>
    </xs:keyref>

```

```

<xs:keyref name="csapat-jatekos" refer="csapatKulcs">
  <xs:selector xpath="jatekosok/jatekos/csapatID"/>
  <xs:field xpath="."/>
</xs:keyref>

<xs:keyref name="jatekos-statisztika" refer="jatekosKulcs">
  <xs:selector xpath="statisztikak/statisztika/jatekosID"/>
  <xs:field xpath="."/>
</xs:keyref>

<xs:keyref name="csapat-jatszik" refer="csapatKulcs">
  <xs:selector xpath="tobbJatszik/jatszik/csapatID"/>
  <xs:field xpath="."/>
</xs:keyref>

<xs:keyref name="meccs-jatszik" refer="meccsKulcs">
  <xs:selector xpath="tobbJatszik/jatszik/meccsID"/>
  <xs:field xpath="."/>
</xs:keyref>

</xs:element>

<xs:complexType name="stadionTipus">
  <xs:sequence>
    <xs:element type="xs:string" name="nev" />
    <xs:element type="xs:string" name="varos" />
    <xs:element type="xs:string" name="cim" />
    <xs:element type="xs:short" name="befogadokepesseg" />
    <xs:element type="xs:short" name="csapatID" />
  </xs:sequence>
  <xs:attribute type="xs:short" name="stadionID" use="required" />
</xs:complexType>

<xs:complexType name="csapatTipus">
  <xs:sequence>
    <xs:element type="xs:string" name="nev" />
    <xs:element type="xs:string" name="varos" />
  </xs:sequence>
  <xs:attribute type="xs:short" name="csapatID" use="required" />
</xs:complexType>

<xs:complexType name="meccsTipus">
  <xs:sequence>
    <xs:element type="xs:string" name="datum" />
    <xs:element type="xs:short" name="stadionID" />
  </xs:sequence>
  <xs:attribute type="xs:short" name="meccsID" use="required" />
</xs:complexType>

```

```
<xs:complexType name="tobbJatszikTipus">
  <xs:sequence>
    <xs:element type="xs:byte" name="hazaiCsapatPontszam" />
    <xs:element type="xs:byte" name="vendegCsapatPontszam" />
    <xs:element type="xs:short" name="csapatID" />
    <xs:element type="xs:short" name="meccsID" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="jatekosTipus">
  <xs:sequence>
    <xs:element type="xs:string" name="nev" />
    <xs:element type="xs:byte" name="kor" />
    <xs:element type="xs:byte" name="mezszám" />
    <xs:element type="xs:string" name="poszt" />
    <xs:element type="xs:short" name="csapatID" />
  </xs:sequence>
  <xs:attribute type="xs:short" name="jatekosID" use="required" />
</xs:complexType>

<xs:complexType name="statisztikaTipus">
  <xs:sequence>
    <xs:element type="xs:int" name="jatszottPerc" />
    <xs:element type="xs:short" name="mezonykosar" />
    <xs:element type="xs:short" name="harompontos" />
    <xs:element type="xs:short" name="bunteto" />
    <xs:element type="xs:short" name="golpassz" />
    <xs:element type="xs:short" name="lepattano" />
    <xs:element type="xs:short" name="jatekosID" />
  </xs:sequence>
  <xs:attribute type="xs:short" name="statisztikaID" use="required" />
</xs:complexType>
</xs:schema>
```

2. feladat

2a) adatolvasás – DomReadU678MF.java

Beolvassa az összes adatot az XMLU678MF.xml fájlból és faszerkezetben kiírja a konzolra.

```
1 package hu.domparsing.U678MF;
2
3 import org.w3c.dom.*;
4 import javax.xml.parsers.*;
5 import java.io.*;
6
7 public class DomReadU678MF {
8     public static void readElements(){
9         try {
10             //Initialize a file
11             File file = new File("XMLU678MF.xml");
12
13             //Create a DocumentBuilder from DocumentBuilderFactory
14             DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
15             DocumentBuilder db = dbf.newDocumentBuilder();
16
17             //Create a Document from a file
18             Document doc = db.parse(file);
19
20             doc.getDocumentElement().normalize();
21
22             //Examine attributes and sub-elements of stadion node
23             NodeList nodeList = doc.getElementsByTagName("stadion");
24             for (int itr = 0; itr < nodeList.getLength(); itr++)
25             {
26                 Node node = nodeList.item(itr);
27                 System.out.println("\n----- " + node.getNodeName() + (itr + 1) + " -----");
28                 if (node.getNodeType() == Node.ELEMENT_NODE)
29                 {
30                     Element eElement = (Element) node;
31                     System.out.println("StadionID: " + eElement.getAttribute("stadionID"));
32                     System.out.println("Nev: " + eElement.getElementsByTagName("nev").item(0).getTextContent());
33                     System.out.println("Varos: " + eElement.getElementsByTagName("varos").item(0).getTextContent());
34                     System.out.println("Cim: " + eElement.getElementsByTagName("cim").item(0).getTextContent());
35                     System.out.println("Befogadokepesseg: " + eElement.getElementsByTagName("befogadokepesseg").item(0).getTextContent());
36                     System.out.println("CsapatID: " + eElement.getElementsByTagName("csapatID").item(0).getTextContent());
37                 }
38             }
39
40             //Examine attributes and sub-elements of csapat node
41             nodeList = doc.getElementsByTagName("csapat");
42             for (int itr = 0; itr < nodeList.getLength(); itr++)
43             {
44                 Node node = nodeList.item(itr);
45                 System.out.println("\n----- " + node.getNodeName() + (itr + 1) + " -----");
46                 if (node.getNodeType() == Node.ELEMENT_NODE)
47                 {
48                     Element eElement = (Element) node;
49                     System.out.println("CsapatID: " + eElement.getAttribute("csapatID"));
50                     System.out.println("Nev: " + eElement.getElementsByTagName("nev").item(0).getTextContent());
51                     System.out.println("Varos: " + eElement.getElementsByTagName("varos").item(0).getTextContent());
52                 }
53             }
54
55             //Examine attributes and sub-elements of meccs node
56             nodeList = doc.getElementsByTagName("meccs");
57             for (int itr = 0; itr < nodeList.getLength(); itr++)
58             {
59                 Node node = nodeList.item(itr);
60                 System.out.println("\n----- " + node.getNodeName() + (itr + 1) + " -----");
61                 if (node.getNodeType() == Node.ELEMENT_NODE)
62                 {
63                     Element eElement = (Element) node;
64                     System.out.println("MeccsID: " + eElement.getAttribute("meccsID"));
65                     System.out.println("Datum: " + eElement.getElementsByTagName("datum").item(0).getTextContent());
66                     System.out.println("StadionID: " + eElement.getElementsByTagName("stadionID").item(0).getTextContent());
67                 }
68             }
69         }
70     }
71 }
```

```

70 //Examine attributes and sub-elements of jatszikk node
71 nodeList = doc.getElementsByTagName("jatszikk");
72 for (int itr = 0; itr < nodeList.getLength(); itr++)
73 {
74     Node node = nodeList.item(itr);
75     System.out.println("\n----- " + node.getNodeName() + (itr + 1) + " -----");
76     if (node.getNodeType() == Node.ELEMENT_NODE)
77     {
78         Element eElement = (Element) node;
79         System.out.println("Hazai csapat pontszám: " + eElement.getElementsByTagName("hazaiCsapatPontszam").item(0).getTextContent());
80         System.out.println("Vendeg csapat pontszám: " + eElement.getElementsByTagName("vendegCsapatPontszam").item(0).getTextContent());
81         System.out.println("CsapatID: " + eElement.getElementsByTagName("csapatID").item(0).getTextContent());
82         System.out.println("MeccsID: " + eElement.getElementsByTagName("meccsID").item(0).getTextContent());
83     }
84 }
85
86 //Examine attributes and sub-elements of jatekos node
87 nodeList = doc.getElementsByTagName("jatekos");
88 for (int itr = 0; itr < nodeList.getLength(); itr++)
89 {
90     Node node = nodeList.item(itr);
91     System.out.println("\n----- " + node.getNodeName() + (itr + 1) + " -----");
92     if (node.getNodeType() == Node.ELEMENT_NODE)
93     {
94         Element eElement = (Element) node;
95         System.out.println("JatekosID: " + eElement.getAttribute("jatekosID"));
96         System.out.println("Nev: " + eElement.getElementsByTagName("nev").item(0).getTextContent());
97         System.out.println("Kor: " + eElement.getElementsByTagName("kor").item(0).getTextContent());
98         System.out.println("Mezszám: " + eElement.getElementsByTagName("mezszám").item(0).getTextContent());
99         System.out.println("Poszt: " + eElement.getElementsByTagName("poszt").item(0).getTextContent());
100         System.out.println("CsapatID: " + eElement.getElementsByTagName("csapatID").item(0).getTextContent());
101     }
102 }
103
104 //Examine attributes and sub-elements of statisztika node
105 nodeList = doc.getElementsByTagName("statisztika");
106 for (int itr = 0; itr < nodeList.getLength(); itr++)
107 {
108     Node node = nodeList.item(itr);
109     System.out.println("\n----- " + node.getNodeName() + (itr + 1) + " -----");
110     if (node.getNodeType() == Node.ELEMENT_NODE)
111     {
112         Element eElement = (Element) node;
113         System.out.println("StatisztikaID: " + eElement.getAttribute("statisztikaID"));
114         System.out.println("Jatszott perc: " + eElement.getElementsByTagName("jatszottPerc").item(0).getTextContent());
115         System.out.println("Mezonykosar: " + eElement.getElementsByTagName("mezonykosar").item(0).getTextContent());
116         System.out.println("Harompontos: " + eElement.getElementsByTagName("harompontos").item(0).getTextContent());
117         System.out.println("Bunteto: " + eElement.getElementsByTagName("bunteto").item(0).getTextContent());
118         System.out.println("Golpassz: " + eElement.getElementsByTagName("golpassz").item(0).getTextContent());
119         System.out.println("Lepattano: " + eElement.getElementsByTagName("lepattano").item(0).getTextContent());
120         System.out.println("JatekosID: " + eElement.getElementsByTagName("jatekosID").item(0).getTextContent());
121     }
122 }
123 }
124 catch (Exception e)
125 {
126     e.printStackTrace();
127 }
128 }
129 }
130

```


Output:

```
----- stadion1 -----
StadionID: 1000
Nev: Staples Center
Varos: Los Angeles
Cim: 1111 S Figueroa St
Befogadokepesseg: 20000
CsapatID: 2000

----- stadion2 -----
StadionID: 1001
Nev: Fiserv Forum
Varos: Milwaukee
Cim: 1111 Vel R. Phillips Ave
Befogadokepesseg: 17341
CsapatID: 2001

----- stadion3 -----
StadionID: 1002
Nev: AmericanAirlines Arena
Varos: Miami
Cim: 601 Biscayne Blvd
Befogadokepesseg: 21000
CsapatID: 2002

----- stadion4 -----
StadionID: 1003
Nev: TD Garden
Varos: Boston
Cim: 100 Legends Way
Befogadokepesseg: 19580
CsapatID: 2003

----- stadion5 -----
StadionID: 1004
Nev: United Center
Varos: Chicago
Cim: 1901 W Madison St
Befogadokepesseg: 23500
CsapatID: 2004

----- stadion6 -----
StadionID: 1005
Nev: Scotiabank Arena
Varos: Toronto
Cim: 40 Bay St
Befogadokepesseg: 19800
CsapatID: 2005

----- stadion7 -----
StadionID: 1006
Nev: AT_T Center
Varos: San Antonio
Cim: 1 AT_T Center Parkway
Befogadokepesseg: 18581
CsapatID: 2006

----- stadion8 -----
StadionID: 1007
Nev: Chase Center
Varos: San Francisco
Cim: 1 Warriors Way
Befogadokepesseg: 18064
CsapatID: 2007
```

```
----- csapat1 -----
CsapatID: 2000
Nev: Lakers
Varos: Los Angeles

----- csapat2 -----
CsapatID: 2001
Nev: Bucks
Varos: Milwaukee

----- csapat3 -----
CsapatID: 2002
Nev: Heat
Varos: Miami

----- csapat4 -----
CsapatID: 2003
Nev: Celtics
Varos: Boston

----- csapat5 -----
CsapatID: 2004
Nev: Bulls
Varos: Chicago

----- csapat6 -----
CsapatID: 2005
Nev: Raptors
Varos: Toronto

----- csapat7 -----
CsapatID: 2006
Nev: Spurs
Varos: San Antonio

----- csapat8 -----
CsapatID: 2007
Nev: Golden State Warriors
Varos: San Francisco

----- meccs1 -----
MeccsID: 3000
Datum: 5/10/2021
StadionID: 1000

----- meccs2 -----
MeccsID: 3001
Datum: 5/18/2021
StadionID: 1002

----- meccs3 -----
MeccsID: 3002
Datum: 6/22/2021
StadionID: 1003

----- meccs4 -----
MeccsID: 3003
Datum: 4/12/2021
StadionID: 1005

----- meccs5 -----
MeccsID: 3004
Datum: 4/9/2021
StadionID: 1004
```

```
----- meccs6 -----
MeccsID: 3005
Datum: 5/3/2021
StadionID: 1001

----- meccs7 -----
MeccsID: 3006
Datum: 6/5/2021
StadionID: 1005

----- meccs8 -----
MeccsID: 3007
Datum: 6/2/2021
StadionID: 1006

----- jatszik1 -----
Hazai csapat pontszam: 113
Vendeg csapat pontszam: 96
CsapatID: 2000
MeccsID: 3000

----- jatszik2 -----
Hazai csapat pontszam: 113
Vendeg csapat pontszam: 96
CsapatID: 2001
MeccsID: 3000

----- jatszik3 -----
Hazai csapat pontszam: 108
Vendeg csapat pontszam: 106
CsapatID: 2002
MeccsID: 3001

----- jatszik4 -----
Hazai csapat pontszam: 108
Vendeg csapat pontszam: 106
CsapatID: 2006
MeccsID: 3001

----- jatszik5 -----
Hazai csapat pontszam: 101
Vendeg csapat pontszam: 101
CsapatID: 2003
MeccsID: 3002

----- jatszik6 -----
Hazai csapat pontszam: 101
Vendeg csapat pontszam: 101
CsapatID: 2004
MeccsID: 3002

----- jatszik7 -----
Hazai csapat pontszam: 87
Vendeg csapat pontszam: 110
CsapatID: 2005
MeccsID: 3003

----- jatszik8 -----
Hazai csapat pontszam: 87
Vendeg csapat pontszam: 110
CsapatID: 2007
MeccsID: 3003

----- jatszik9 -----
Hazai csapat pontszam: 103
Vendeg csapat pontszam: 105
CsapatID: 2004
MeccsID: 3004
```

```
----- jatszik10 -----
Hazai csapat pontszám: 103
Vendég csapat pontszám: 105
CsapatID: 2002
MeccsID: 3004

----- jatekos1 -----
JatekosID: 4000
Nev: LeBron James
Kor: 36
Mezszám: 23
Poszt: SF
CsapatID: 2000

----- jatekos2 -----
JatekosID: 4001
Nev: Giannis Antetokounmpo
Kor: 26
Mezszám: 34
Poszt: PF
CsapatID: 2001

----- jatekos3 -----
JatekosID: 4002
Nev: Jimmy Butler
Kor: 32
Mezszám: 22
Poszt: SG/SF
CsapatID: 2002

----- jatekos4 -----
JatekosID: 4003
Nev: Jaylen Brown
Kor: 25
Mezszám: 7
Poszt: SG/SF
CsapatID: 2003

----- jatekos5 -----
JatekosID: 4004
Nev: Zach LaVine
Kor: 26
Mezszám: 24
Poszt: SG/SF
CsapatID: 2004

----- jatekos6 -----
JatekosID: 4005
Nev: Kyle Lowry
Kor: 35
Mezszám: 7
Poszt: PG
CsapatID: 2005

----- jatekos7 -----
JatekosID: 4006
Nev: DeMar DeRozan
Kor: 32
Mezszám: 10
Poszt: SF/SG
CsapatID: 2006

----- jatekos8 -----
JatekosID: 4007
Nev: DAngelo Russel
Kor: 25
Mezszám: 0
Poszt: PG
CsapatID: 2007
```

```
----- statisztika1 -----
StatisztikaID: 5000
Jatszott perc: 20102
Mezonykosar: 3841
Harompontos: 2103
Bunteto: 3421
Golpassz: 1643
Lepattano: 1675
JatekosID: 4000

----- statisztika2 -----
StatisztikaID: 5001
Jatszott perc: 62231
Mezonykosar: 4232
Harompontos: 4622
Bunteto: 5456
Golpassz: 2447
Lepattano: 3452
JatekosID: 4001

----- statisztika3 -----
StatisztikaID: 5002
Jatszott perc: 32131
Mezonykosar: 1265
Harompontos: 4732
Bunteto: 5321
Golpassz: 5644
Lepattano: 3534
JatekosID: 4002

----- statisztika4 -----
StatisztikaID: 5003
Jatszott perc: 21333
Mezonykosar: 4624
Harompontos: 2458
Bunteto: 3215
Golpassz: 5464
Lepattano: 3511
JatekosID: 4003

----- statisztika5 -----
StatisztikaID: 5004
Jatszott perc: 13865
Mezonykosar: 9754
Harompontos: 5564
Bunteto: 6845
Golpassz: 6454
Lepattano: 2135
JatekosID: 4004

----- statisztika6 -----
StatisztikaID: 5005
Jatszott perc: 78695
Mezonykosar: 1321
Harompontos: 5468
Bunteto: 5448
Golpassz: 5464
Lepattano: 3543
JatekosID: 4005

----- statisztika7 -----
StatisztikaID: 5006
Jatszott perc: 21515
Mezonykosar: 4648
Harompontos: 3213
Bunteto: 2135
Golpassz: 4421
Lepattano: 5443
JatekosID: 4006
```

```
----- statisztika8 -----
StatisztikaID: 5007
Jatszott perc: 94656
Mezonykosar: 6521
Harompontos: 5462
Bunteto: 5468
Golpassz: 1353
Lepattano: 1235
JatekosID: 4007
```

2b) adatmódosítás – DomModifyU678MF.java

Bekér adatokat a felhasználótól hibaellenőrzéssel, majd módosítja az XMLU678MF.xml fájl kívánt elemét a megadott értékre. Do-while ciklusban fut, így több adat módosítására is van lehetőség. Ezután az eredményt kiírja a konzolra és elmenti egy új modifiedXml.xml fájlba (úgy van konfigurálva, hogy nem az eredeti fájl módosul).

```
1 package hu.domparsing.U678MF;
2
3 import org.w3c.dom.*;
4 import org.xml.sax.SAXException;
5
6 import javax.xml.parsers.*;
7 import javax.xml.transform.Transformer;
8 import javax.xml.transform.TransformerFactory;
9 import javax.xml.transform.dom.DOMSource;
10 import javax.xml.transform.stream.StreamResult;
11
12 import java.io.*;
13 import java.util.ArrayList;
14 import java.util.List;
15
16 public class DomModifyU678MF {
17     //Instantiate a BufferedReader for read user input
18     static BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
19
20     //Method to
21     //Create a DocumentBuilder from DocumentBuilderFactory
22     //Create a Document from a file
23     //Do the updating of elements as long as the user wants
24     //Call the printing method in the end to print out the results and save it to a new file
25     public static void modifyElements() {
26         try {
27             DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
28             DocumentBuilder db = dbf.newDocumentBuilder();
29             Document dom = db.parse(new File("XMLU678MF.xml"));
30
31             Document modifiedDocument = dom;
32             boolean yes = true;
33
34             do {
35                 modifiedDocument = update(modifiedDocument);
36                 System.out.println("Szeretnél további elemeket módosítani?\nHa igen, akkor nyomj 1-et, ha nem akkor bármi mást.");
37                 if(reader.readLine().compareTo("1") != 0) {
38                     yes = false;
39                 }
40             } while(yes);
41
42             print(modifiedDocument);
43
44         } catch (ParserConfigurationException pce) {
45             pce.printStackTrace();
46         } catch (SAXException se) {
47             se.printStackTrace();
48         } catch (Exception ioe) {
49             System.err.println("kúuutya");
50             ioe.printStackTrace();
51         }
52     }
53 }
```

```

55 //Method to do the actual updating of an element based on user-specified parameters
56 public static Document update(Document dom) {
57     Node rootNode = dom.getDocumentElement();
58     Node innerNode = null;
59     Node inner2Node = null;
60     Node inner3Node = null;
61
62     List<String> parameters = null;
63     boolean ok = false;
64
65     while(!ok) {
66         try {
67             parameters = getDataFromUser();
68             ok = true;
69         } catch (Exception e) {
70             System.out.println("Hibás azonosító! Fussunk neki megegyszer!\n");
71         }
72     }
73
74
75     for (int i = 0; i < rootNode.getChildNodes().getLength(); i++) {
76         innerNode = rootNode.getChildNodes().item(i);
77
78         if(innerNode.getNodeType() == Node.ELEMENT_NODE) {
79             for (int j = 0; j < innerNode.getChildNodes().getLength(); j++) {
80                 inner2Node = innerNode.getChildNodes().item(j);
81
82                 if(inner2Node.getNodeType() == Node.ELEMENT_NODE) {
83                     Element e1 = (Element) inner2Node;
84
85                     if(parameters.get(0).compareTo(inner2Node.getNodeName()) == 0) {
86                         if(parameters.get(2).compareTo(e1.getAttribute(parameters.get(1))) == 0) {
87                             for (int k = 0; k < e1.getChildNodes().getLength(); k++) {
88                                 inner3Node = e1.getChildNodes().item(k);
89
90                                 if(inner3Node.getNodeType() == Node.ELEMENT_NODE) {
91                                     Element e2 = (Element) inner3Node;
92
93                                     if(e2.getNodeName().compareTo(parameters.get(3)) == 0) {
94                                         e2.setTextContent(parameters.get(4));
95                                     }
96                                 }
97                             }
98                         }
99                     }
100                 }
101             }
102         }
103     }
104
105     return dom;
106 }
107

```

```

108 //Method to get data from user
109 public static List<String> getDataFromUser() throws IOException {
110     List<String> resultList = new ArrayList<String>();
111
112     System.out.println("Add meg a kódját, hogy melyik node-on szeretnél módosítani:");
113     System.out.println("stadion - 1\ncsapat - 2\njatekos - 3\nstatisztika - 4");
114
115     //Reading data from user
116     String codeString = reader.readLine();
117
118     if(codeString.compareTo("1") == 0 || codeString.compareTo("2") == 0 || codeString.compareTo("3") == 0 || codeString.compareTo("4") == 0) {
119
120         List<String> x = null;
121         boolean ok = false;
122
123         switch (codeString) {
124             case "1":
125                 resultList.add("stadion");
126                 resultList.add("stadionID");
127
128                 while(!ok) {
129                     try {
130                         x = stadionMethod();
131                         ok = true;
132                     } catch (IllegalArgumentException e) {
133                         System.out.println("Nem ervenyes adat! Lássuk csak még egyszer!\n");
134                     } catch (IOException e) {
135                         System.out.println("Nem letezo stadionID! Add meg újra!");
136                     }
137                 }
138
139                 resultList.add(x.get(0));
140                 resultList.add(x.get(1));
141                 resultList.add(x.get(2));
142                 break;
143             case "2":
144                 System.out.println("kkkkiskutya");
145                 resultList.add("csapat");
146                 resultList.add("csapatID");
147
148                 while(!ok) {
149                     try {
150                         x = csapatMethod();
151                         ok = true;
152                     } catch (IllegalArgumentException e) {
153                         System.out.println("Nem ervenyes adat! Lássuk csak még egyszer!\n");
154                     } catch (IOException e) {
155                         System.out.println("Nem letezo csapatID! Add meg újra!");
156                     }
157                 }
158
159                 resultList.add(x.get(0));
160                 resultList.add(x.get(1));
161                 resultList.add(x.get(2));
162                 break;
163             case "3":
164                 resultList.add("jatekos");
165                 resultList.add("jatekosID");
166
167                 while(!ok) {
168                     try {
169                         x = jatekosMethod();
170                         ok = true;
171                     } catch (IllegalArgumentException e) {
172                         System.out.println("Nem ervenyes adat! Lássuk csak még egyszer!\n");
173                     } catch (IOException e) {
174                         System.out.println("Nem letezo jatekosID! Add meg újra!");
175                     }
176                 }
177

```

```

178         resultList.add(x.get(0));
179         resultList.add(x.get(1));
180         resultList.add(x.get(2));
181         break;
182     case "4":
183         resultList.add("statisztika");
184         resultList.add("statisztikaID");
185
186         while(!ok) {
187             try {
188                 x = statisztikaMethod();
189                 ok = true;
190             } catch (IllegalArgumentException e) {
191                 System.out.println("Nem ervenyes adat! Lássuk csak még egyszer!\n");
192             } catch (IOException e) {
193                 System.out.println("Nem letezo statisztikaID! Add meg újra!");
194             }
195         }
196
197         resultList.add(x.get(0));
198         resultList.add(x.get(1));
199         resultList.add(x.get(2));
200         break;
201     }
202 } else {
203     throw new IllegalArgumentException("Unexpected value: " + codeString);
204 }
205
206 return resultList;
207 }
208
209 //Method to handle stadion node modifications
210 private static List<String> stadionMethod() throws IOException {
211     List<String> resultList = new ArrayList<String>();
212
213
214     System.out.println("Add meg a módosítani kívánt Stadion ID-ját:");
215     String idString = reader.readLine();
216
217     if(Integer.parseInt(idString) < 1000 || Integer.parseInt(idString) > 1007) {
218         throw new IOException();
219     } else {
220         resultList.add(idString);
221     }
222
223     System.out.println("Add meg a kódját, hogy a StadionID: " + idString + " stadion melyik node-ján szeretnél módosítani:");
224     System.out.println("nev - 1\nbefogadokepesseg - 2\ncsapatID - 3");
225
226     String code = reader.readLine();
227
228     if(code.compareTo("1") == 0) {
229         resultList.add("nev");
230         System.out.println("Add meg a stadion új nevét: ");
231         resultList.add(reader.readLine());
232     }
233     else if (code.compareTo("2") == 0) {
234         resultList.add("befogadokepesseg");
235         System.out.println("Add meg a stadion új befogadokepesseget: ");
236         resultList.add(reader.readLine());
237     }
238     else if (code.compareTo("3") == 0) {
239         resultList.add("csapatID");
240         System.out.println("Add meg a stadion új csapatID-ját: ");
241         resultList.add(reader.readLine());
242     }
243     else {
244         throw new IllegalArgumentException("Unexpected value: " + code);
245     }
246
247     return resultList;
248 }

```

```

250 //Method to handle csapat node modifications
251 private static List<String> csapatMethod() throws IOException {
252     List<String> resultList = new ArrayList<String>();
253
254     BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
255
256     System.out.println("Add meg a módosítani kívánt Csapat ID-ját:");
257     String idString = reader.readLine();
258
259     if(Integer.parseInt(idString) < 2000 || Integer.parseInt(idString) > 2007) {
260         throw new IOException();
261     } else {
262         resultList.add(idString);
263     }
264
265     System.out.println("Add meg a kódját, hogy a CsapatID: " + idString + " csapat melyik node-ján szeretnél módosítani:");
266     System.out.println("nev - 1\nvaros - 2");
267
268     String code = reader.readLine();
269
270     if(code.compareTo("1") == 0) {
271         resultList.add("nev");
272         System.out.println("Add meg a csapat új nevét: ");
273         resultList.add(reader.readLine());
274
275     } else if (code.compareTo("2") == 0) {
276         resultList.add("varos");
277         System.out.println("Add meg a csapat új városát: ");
278         resultList.add(reader.readLine());
279
280     } else {
281         throw new IllegalArgumentException("Unexpected value: " + code);
282     }
283
284     return resultList;
285 }
286
287 //Method to handle jatekos node modifications
288 private static List<String> jatekosMethod() throws IOException {
289     List<String> resultList = new ArrayList<String>();
290
291     BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
292
293     System.out.println("Add meg a módosítani kívánt Jatekos ID-ját:");
294     String idString = reader.readLine();
295
296     if(Integer.parseInt(idString) < 4000 || Integer.parseInt(idString) > 4007) {
297         throw new IOException();
298     } else {
299         resultList.add(idString);
300     }
301
302     System.out.println("Add meg a kódját, hogy a JatekosID: " + idString + " jatekos melyik node-ján szeretnél módosítani:");
303     System.out.println("nev - 1\nmezsza - 2\nposzt - 3\n csapatid - 4");
304
305     String code = reader.readLine();
306
307     if(code.compareTo("1") == 0) {
308         resultList.add("nev");
309         System.out.println("Add meg a jatekos új nevét: ");
310         resultList.add(reader.readLine());
311
312     } else if (code.compareTo("2") == 0) {
313         resultList.add("mezsza");
314         System.out.println("Add meg a jatekos új mezszaát: ");
315         resultList.add(reader.readLine());
316
317     } else if (code.compareTo("3") == 0) {
318         resultList.add("poszt");
319         System.out.println("Add meg a jatekos új posztját: ");
320         resultList.add(reader.readLine());

```

```

322     } else if (code.compareTo("4") == 0) {
323         resultList.add("csapatID");
324         System.out.println("Add meg a jatekos új csapatID-ját: ");
325         resultList.add(reader.readLine());
326
327     } else {
328         throw new IllegalArgumentException("Unexpected value: " + code);
329     }
330
331     return resultList;
332 }
333
334 //Method to handle statisztika node modifications
335 private static List<String> statisztikaMethod() throws IOException {
336     List<String> resultList = new ArrayList<String>();
337
338     BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
339
340     System.out.println("Add meg a módosítani kívánt Statisztika ID-ját:");
341     String idString = reader.readLine();
342
343     if(Integer.parseInt(idString) < 5000 || Integer.parseInt(idString) > 5007) {
344         throw new IOException();
345     } else {
346         resultList.add(idString);
347     }
348
349     System.out.println("Add meg a kódját, hogy a StatisztikaID: " + idString + " statisztika melyik node-ján szeretnél módosítani:");
350     System.out.println("mezonkosar - 1\nharompontos - 2\nbunteto - 3\ncgolpassz - 4\nlepattano - 5");
351
352     String code = reader.readLine();
353
354     if(code.compareTo("1") == 0) {
355         resultList.add("mezonkosar");
356         System.out.println("Add meg a jatekos statisztika új mezonkosarainak szamat: ");
357         resultList.add(reader.readLine());
358
359     } else if (code.compareTo("2") == 0) {
360         resultList.add("harompontos");
361         System.out.println("Add meg a jatekos statisztika új harompontosainak szamat: ");
362         resultList.add(reader.readLine());
363
364     } else if (code.compareTo("3") == 0) {
365         resultList.add("bunteto");
366         System.out.println("Add meg a jatekos statisztika új buntetoinek szamat: ");
367         resultList.add(reader.readLine());
368
369     } else if (code.compareTo("4") == 0) {
370         resultList.add("cgolpassz");
371         System.out.println("Add meg a jatekos statisztika új cgolpasszainak szamat: ");
372         resultList.add(reader.readLine());
373
374     } else if (code.compareTo("5") == 0) {
375         resultList.add("lepattano");
376         System.out.println("Add meg a jatekos statisztika új lepattanoinak szamat: ");
377         resultList.add(reader.readLine());
378
379     } else {
380         throw new IllegalArgumentException("Unexpected value: " + code);
381     }
382
383     return resultList;
384 }
385

```

```

386 //Method to print the modified xml to the console and save it to a new file
387 public static void print(Document doc) {
388     try {
389         TransformerFactory transformerFactory = TransformerFactory.newInstance();
390         Transformer transformer = transformerFactory.newTransformer();
391         DOMSource domSource = new DOMSource(doc);
392
393         StreamResult console = new StreamResult(System.out);
394         StreamResult file = new StreamResult(new File("modifiedXml.xml"));
395
396         transformer.transform(domSource, console);
397         transformer.transform(domSource, file);
398     } catch (Exception e) {
399         e.printStackTrace();
400     }
401 }
402 }
403 }

```


Output:

```
Add meg a kódját, hogy melyik node-on szeretnél módosítani:
stadion - 1
csapat - 2
jatekos - 3
statisztika - 4
1
Add meg a módosítani kívánt Stadion ID-ját:
1005
Add meg a kódját, hogy a StadionID: 1005 stadion melyik node-ján szeretnél módosítani:
nev - 1
befogadokepesseg - 2
csapatID - 3
2
Add meg a stadion új befogadokepesseget:
19870
Szeretnél további elemeket módosítani?
Ha igen, akkor nyomj 1-et, ha nem akkor bármi más.
n
<?xml version="1.0" encoding="UTF-8" standalone="no"?><nba>
  <stadionok>
    <stadion stadionID="1000">
      <nev>Staples Center</nev>
      <varos>Los Angeles</varos>
      <cim>1111 S Figueroa St</cim>
      <befogadokepesseg>20000</befogadokepesseg>
      <csapatID>2000</csapatID>
    </stadion>
    <stadion stadionID="1001">
      <nev>Fiserv Forum</nev>
      <varos>Milwaukee</varos>
      <cim>1111 Vel R. Phillips Ave</cim>
      <befogadokepesseg>17341</befogadokepesseg>
      <csapatID>2001</csapatID>
    </stadion>
    <stadion stadionID="1002">
      <nev>AmericanAirlines Arena</nev>
      <varos>Miami</varos>
      <cim>601 Biscayne Blvd</cim>
      <befogadokepesseg>21000</befogadokepesseg>
      <csapatID>2002</csapatID>
    </stadion>
    <stadion stadionID="1003">
      <nev>TD Garden</nev>
      <varos>Boston</varos>
      <cim>100 Legends Way</cim>
      <befogadokepesseg>19580</befogadokepesseg>
      <csapatID>2003</csapatID>
    </stadion>
    <stadion stadionID="1004">
      <nev>United Center</nev>
      <varos>Chicago</varos>
      <cim>1901 W Madison St</cim>
      <befogadokepesseg>23500</befogadokepesseg>
      <csapatID>2004</csapatID>
    </stadion>
    <stadion stadionID="1005">
      <nev>Scotiabank Arena</nev>
      <varos>Toronto</varos>
      <cim>40 Bay St</cim>
      <befogadokepesseg>19870</befogadokepesseg>
      <csapatID>2005</csapatID>
    </stadion>
    <stadion stadionID="1006">
      <nev>AT_T Center</nev>
      <varos>San Antonio</varos>
      <cim>1 AT_T Center Parkway</cim>
      <befogadokepesseg>18581</befogadokepesseg>
      <csapatID>2006</csapatID>
    </stadion>
    <stadion stadionID="1007">
      <nev>Chase Center</nev>
```

3c) adatlekérdezés – DomQueryU678MF.java

3 db adatlekérdezést valósít meg:

- 20 000 főnél nagyobb befogadóképességű stadionok adatai
- A 4006-os játékosID-val rendelkező játékos adatai
- A 3000-nél több, de 5600-nál kevesebb golpasszal rendelkező statisztikák adatai

```
1 package hu.domparse.U678MF;
2
3 import org.w3c.dom.*;
4 import javax.xml.parsers.*;
5 import javax.xml.xpath.XPath;
6 import javax.xml.xpath.XPathFactory;
7 import javax.xml.xpath.XPathConstants;
8
9 import java.io.*;
10
11 public class DomQueryU678MF {
12     public static void queryElements() {
13         try {
14             //Initialize a file
15             File inputFile = new File("XMLU678MF.xml");
16
17             //Create a DocumentBuilder from DocumentBuilderFactory
18             DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
19             DocumentBuilder db = dbf.newDocumentBuilder();
20
21             //Create a Document from a file
22             Document doc = db.parse(inputFile);
23
24             doc.getDocumentElement().normalize();
25
26             // Create XPath object
27             XPath xPath = XPathFactory.newInstance().newXPath();
28
29             //-----QUERIES-----
30
31             // Queries stadiums with a capacity of more than 20 000
32             String query1 = "nba/stadionok/stadion[befogadokepesszeg>20000]";
33
34             // Queries the player with jatekosID 4006
35             String query2 = "nba/jatekosok/jatekos[@jatekosID='4006']";
36
37             // Queries statistics where the number of assists is more than 3000 and less than 5600
38             String query3 = "nba/statisztikak/statisztika[golpassz>3000 and golpassz<5600]";
39
40
41             //-----
42
43             // Evaluation of query expression
44             NodeList nodeList = (NodeList) xPath.compile(query1).evaluate(doc, XPathConstants.NODESET);
45             System.out.println("Query result of stadiums with a capacity of more than 20 000\n");
46             printElements(nodeList, "stadionID");
47
48             System.out.println("\n-----\n");
49
50             // Evaluation of query expression
51             nodeList = (NodeList) xPath.compile(query2).evaluate(doc, XPathConstants.NODESET);
52             System.out.println("Query result of player with jatekosID 4006 \n");
53             printElements(nodeList, "jatekosID");
54
55             System.out.println("\n-----\n");
56
57             // Evaluation of query expression
58             nodeList = (NodeList) xPath.compile(query3).evaluate(doc, XPathConstants.NODESET);
59             System.out.println("Query result of statistics where the number of assists is more than 3000 and less than 5600\n");
60             printElements(nodeList, "statisztikaID");
61
62         } catch (Exception e) {
63             System.out.println("Some error ocurred\nDescription:\n" + e.getMessage());
64             e.printStackTrace();
65         }
66     }
67 }
```

```

68 //Method to print the elements of a NodeList formatted
69 private static void printElements(NodeList nodeList, String idString) {
70     for (int i = 0; i < nodeList.getLength(); i++) {
71         Node node = nodeList.item(i);
72         System.out.println("Elem: " + node.getNodeName());
73         Node nNode = nodeList.item(i);
74         if (nNode.getNodeType() == Node.ELEMENT_NODE) {
75             Element elem = (Element)nNode;
76
77             // Print the identifier
78             String id = elem.getAttribute(idString);
79             System.out.println(idString + ": " + id);
80
81             // Print child nodes
82             String nodeContent = "";
83             NodeList childNodes = elem.getChildNodes();
84             for (int j = 0; j < childNodes.getLength(); j++) {
85                 if (childNodes.item(j).getTextContent().trim() != "") {
86                     nodeContent = formatText(childNodes.item(j).getTextContent().trim());
87                     System.out.println(childNodes.item(j).getNodeName() + ": " + nodeContent);
88                 }
89             }
90         }
91         System.out.println();
92     }
93 }
94
95 // Method to format text for printing
96 private static String formatText(String text) {
97     text = text.replaceAll("\\n", ", ");
98     text = text.replaceAll("\\s+", " ");
99     return text;
100 }
101 }

```

Output:

Query result of stadiums with a capacity of more than 20 000

Elem: stadion
stadionID: 1002
nev: AmericanAirlines Arena
varos: Miami
cim: 601 Biscayne Blvd
befogadokepesseg: 21000
csapatID: 2002

Elem: stadion
stadionID: 1004
nev: United Center
varos: Chicago
cim: 1901 W Madison St
befogadokepesseg: 23500
csapatID: 2004

Query result of player with jatekosID 4006

Elem: jatekos
jatekosID: 4006
nev: DeMar DeRozan
kor: 32
mezzsam: 10
poszt: SF/SG
csapatID: 2006

Query result of statistics where the number of assists is more than 3000 and less than 5600

Elem: statisztika
statisztikaID: 5003
jatszottPerc: 21333
mezonykosar: 4624
harompontos: 2458
bunteto: 3215
golpassz: 5464
lepattano: 3511
jatekosID: 4003

Elem: statisztika
statisztikaID: 5005
jatszottPerc: 78695
mezonykosar: 1321
harompontos: 5468
bunteto: 5448
golpassz: 5464
lepattano: 3543
jatekosID: 4005

Elem: statisztika
statisztikaID: 5006
jatszottPerc: 21515
mezonykosar: 4648
harompontos: 3213
bunteto: 2135
golpassz: 4421
lepattano: 5443
jatekosID: 4006