

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Készítette: **Horanek Dávid**

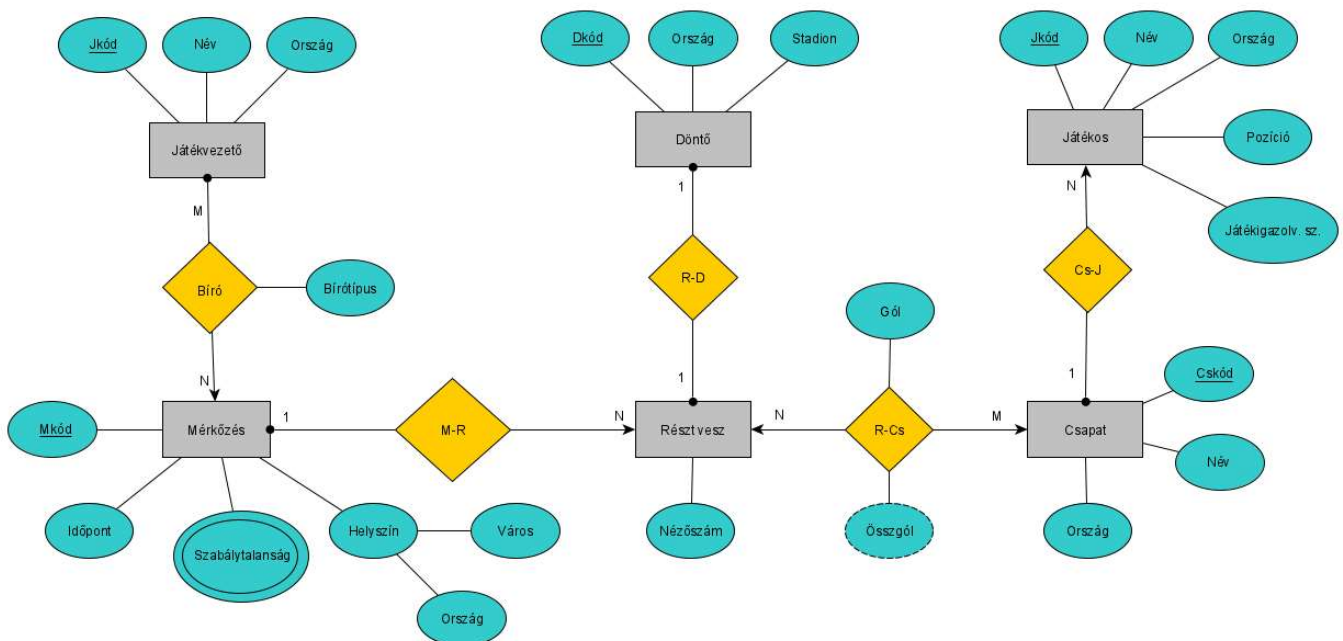
Neptunkód: **L3N7UO**

A feladat leírása:

A feladatban egy futball mérkőzésről készítettem egy XML fájlt. Az XML tartalmazza a mérkőzésen résztvevő csapatok nevét és egyéb adatait, a csapatokban játszó játékosok nevét és további adatait. Ezek mellett a lejátszott mérkőzést vezető játékvezetőket és adatait.

Ez a mérkőzés egy döntő lesz, amelyen két csapat vesz részt melyek a **resztvesz** kapcsolótáblában vannak összekötve. A **resztvesz** tábla tartalmazza a mérkőzésen megjelent nézők számát mely opcionális mező mivel nem mindig lehetnek nézők a mérkőzésen. A **resztvesz** tartalmazza a döntő egyedi azonosítóját is, ami megmondja, hogy hol játszották le a döntőt (ország, stadion). A mérkőzésen szerepelnek játékvezetők is, akik a **biro** kapcsolótábla segítségével vannak hozzákötve a **merkozes**-hez. A **biro** rendelkezik egy **tipus** mezővel amelyikben megmondjuk a játékvezetőnek a mérkőzésen betöltött szerepét. A mérkőzés során lehetnek olyan szabálytalanságok, amelyek után lapot kaphatnak a játékosok ezért a **szabalytalansag**-ban írjuk fel ezeket. A csapatokat az **r-cs** kapcsolótáblán keresztül vannak összekötve az ahhoz a mérkőzéshez, amin részt vesznek. A **resztvesz** és a csapat egyedi azonosítóját tartalmazza. Ezen kívül tartalmaz egy **gol** elemet, ami a csapatnak a mérkőzésen rúgott góljainak számát tárolja. A mérkőzésen szereplő csapatok játékosai a **cs-j** kapcsolótáblán keresztül kerülnek összekötésre a saját csapataikkal. Ide a csapat és a játékos egyedi azonosítója kerül be. A játékosok adatainak felvételekor a játékosigazolvány szám megadása kötelező mezőként szerepel.

Adatbázis ER modellje:



Az adatbázis konvertálása XDM modellre:

A gyökér elemnek a `document`-et használtam és ez alá rendeződik be minden. Mivel a játékvezetők, csapatok, valamint játékosok (kapcsolótáblájukból) is több lehet ezért létrehoztam hozzájuk egy csoportosító mezőt, hogy az XML-ben szebben nézzen ki.

Játékvezetők elem:

Többszörösen előforduló elem. Gyerek elemei rendelkeznek egy attribútummal, ami az egyedi azonosítója, valamint két elemmel.

Bírók elem:

Többszörösen előforduló elem. Összekapcsolja a mérkőzést a bírókkal, rendelkezik két attribútummal, ami mindkettő idegen kulcs, a játékvezető és a mérkőzés egyedi azonosítója.

Mérkőzés:

Rendelkezik egy attribútummal, ami az egyedi azonosítója, egy időpont elemmel, és egy helyszín elemmel, aminek a gyerek elemei az ország és a város. Ezen kívül van egy többszörösen előforduló elem a szabálytalanságok.

Döntő elem:

Rendelkezik egy attribútummal, ami az egyedi azonosítója, egy ország és egy stadion elemmel.

Résztesz:

Rendelkezik három attribútummal, egy egyedi azonosító és két idegen kulccsal. Az idegen kulcsok a döntő és a mérkőzés egyedi azonosítói. Valamint van egy opcionális eleme.

Résztesz-Csapatok(r-cs):

Többszörösen előforduló elem. Rendelkezik két attribútummal, mindkettő idegen kulcs, ami a résztvesz és a csapatok egyedi azonosítói. Tartalmaz egy gól elemet.

Csapatok:

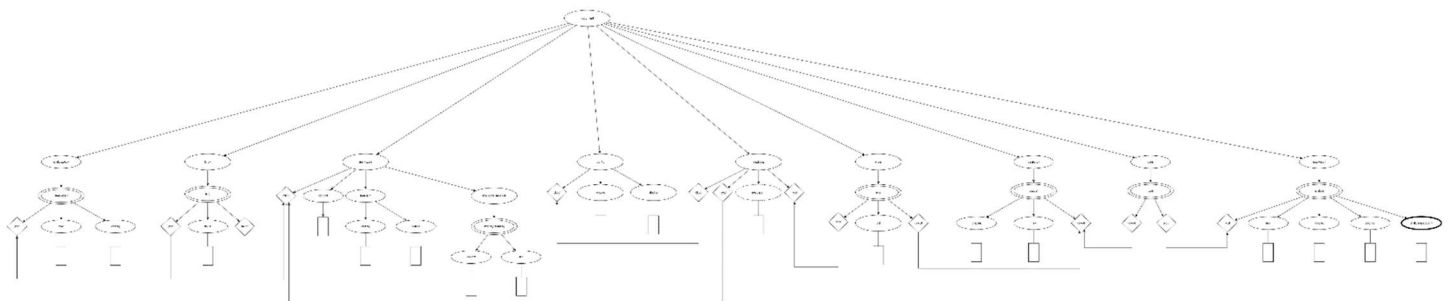
Többszörösen előforduló elem. Tartalmaz egy attribútumot, ami az egyedi azonosítója, valamint két elemet.

Csapatok-Játékosok(cs-j):

Többszörösen előforduló elem. Két attribútummal rendelkezik, mindkettő idegen kulcs, a csapatok és kátekosok egyedi azonosítói.

Játékosok:

Többszörösen előforduló elem. Rendelkezik egy attribútummal, ami az egyedi azonosítója, négy elemmel melyek közül egy kötelező.



XML dokumentum kódja:

```
<?xml version="1.0" encoding="UTF-8"?>
<document xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
xsi:noNamespaceSchemaLocation="XMLSchemaL3N7U0.xsd">

  <!-- ER modell merkozes-jatekvezeto része(bal oldal) -->
  <merkozes mkod="M01">
    <idopont>2022-06-23</idopont>
    <helyszin>
      <orszag>Oroszország</orszag>
      <varos>Moszkva</varos>
    </helyszin>
    <szabalytalansagok>
      <szabalytalansag mkod="M01">
        <idopont>37</idopont>
        <lap>Sarga</lap>
      </szabalytalansag>

      <szabalytalansag mkod="M01">
        <idopont>65</idopont>
        <lap>Piros</lap>
      </szabalytalansag>
    </szabalytalansagok>
  </merkozes>

  <jatekvezetok>
    <jatekvezeto jkod="B01">
      <nev>Smith Jhon</nev>
      <orszag>Anglia</orszag>
    </jatekvezeto>
    <jatekvezeto jkod="B02">
      <nev>Dany Lewis</nev>
      <orszag>Skócia</orszag>
    </jatekvezeto>
    <jatekvezeto jkod="B03">
      <nev>Big Pete</nev>
      <orszag>Anglia</orszag>
    </jatekvezeto>
  </jatekvezetok>

  <birok>
    <biro mkod="M01" jkod="B01">
      <birotipus>Jatekvezeto</birotipus>
    </biro>
    <biro mkod="M01" jkod="B02">
      <birotipus>Partjelzo</birotipus>
    </biro>
    <biro mkod="M01" jkod="B03">
      <birotipus>Videobiro</birotipus>
    </biro>
  </birok>

  <!-- ER modell jatekos-csapat része(jobb oldal) -->
  <jatekosok>
    <jatekos jkod="J01">
      <nev>Karim Benzema</nev>
      <orszag>Franciaország</orszag>
      <pozicio>Tamado</pozicio>
      <jatekosigazolvany>ABC123</jatekosigazolvany>
    </jatekos>
    <jatekos jkod="J02">
      <nev>Toni Kroos</nev>
      <orszag>Németország</orszag>
      <pozicio>Kozepallyas</pozicio>
      <jatekosigazolvany>ABC456</jatekosigazolvany>
    </jatekos>
    <jatekos jkod="J03">
      <nev>Thibaut Courtois</nev>
      <orszag>Belgium</orszag>
      <pozicio>Kapus</pozicio>
      <jatekosigazolvany>ABC789</jatekosigazolvany>
    </jatekos>
  </jatekosok>
</document>
```

```

    <jatekos jkod="J04">
      <nev>Robert Lewandowski</nev>
      <orszag>Lengyelország</orszag>
      <pozicio>Tamado</pozicio>
      <jatekosigazolvany>CBA123</jatekosigazolvany>
    </jatekos>
    <jatekos jkod="J05">
      <nev>Leon Goretzka</nev>
      <orszag>Németország</orszag>
      <pozicio>Középpályás</pozicio>
      <jatekosigazolvany>CBA456</jatekosigazolvany>
    </jatekos>
    <jatekos jkod="J06">
      <nev>Manuel Neuer</nev>
      <orszag>Németország</orszag>
      <pozicio>Kapus</pozicio>
      <jatekosigazolvany>CBA789</jatekosigazolvany>
    </jatekos>
  </jatekosok>

  <csapatok>
    <csapat cskod="CS01">
      <nev>Real Madrid</nev>
      <orszag>Spanyolország</orszag>
    </csapat>
    <csapat cskod="CS02">
      <nev>Bayern München</nev>
      <orszag>Németország</orszag>
    </csapat>
  </csapatok>

  <cs-js>
    <cs-j jkod="J01" cskod="CS01"/>
    <cs-j jkod="J02" cskod="CS01"/>
    <cs-j jkod="J03" cskod="CS01"/>
    <cs-j jkod="J04" cskod="CS02"/>
    <cs-j jkod="J05" cskod="CS02"/>
    <cs-j jkod="J06" cskod="CS02"/>
  </cs-js>

  <!-- ER modell résztvevő része(közep) -->
  <donto dkod="D01">
    <orszag>Oroszország</orszag>
    <stadion>Saint Petersburg Stadium</stadion>
  </donto>

  <resztvesz rkod="R01" mkod="M01" dkod="D01">
    <nezoszam>10000000</nezoszam>
  </resztvesz>

  <r-css>
    <r-cs rkod="R01" cskod="CS01">
      <gol>5</gol>
    </r-cs>
    <r-cs rkod="R01" cskod="CS02">
      <gol>3</gol>
    </r-cs>
  </r-css>
</document>

```

XMLSchema kódja:

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

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">

  <!-- XML fő részei saját típussal -->
  <xs:element name="document">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="merkozes" type="merkozesTipus"/>
        <xs:element name="jatekvezetok" type="jatekvezetokTipus"/>
        <xs:element name="birok" type="birokTipus"/>
        <xs:element name="jatekosok" type="jatekosokTipus"/>
        <xs:element name="csapatok" type="csapatokTipus"/>
        <xs:element name="cs-js" type="cs-jsTipus"/>
        <xs:element name="donto" type="dontoTipus"/>
        <xs:element name="resztvesz" type="resztveszTipus"/>
        <xs:element name="r-css" type="r-cssTipus"/>
      </xs:sequence>
    </xs:complexType>
  <!-- Merkozes-Jatekvezeto megszoritas -->
  <xs:key name="keyMerkozes">
    <xs:selector xpath="merkozes"/>
    <xs:field xpath="@mkod"/>
  </xs:key>
  <xs:key name="keyJatekvezeto">
    <xs:selector xpath="jatekvezetok/jatekvezeto"/>
    <xs:field xpath="@jkod"/>
  </xs:key>

  <xs:keyref name="keyrefBiro1" refer="keyMerkozes">
    <xs:selector xpath="birok/biro"/>
    <xs:field xpath="@mkod"/>
  </xs:keyref>
  <xs:keyref name="keyrefBiro2" refer="keyJatekvezeto">
    <xs:selector xpath="birok/biro"/>
    <xs:field xpath="@jkod"/>
  </xs:keyref>

  <!-- Jatekos-Csapat megszoritas -->
  <xs:key name="keyJatekos">
    <xs:selector xpath="jatekosok/jatekos"/>
    <xs:field xpath="@jkod"/>
  </xs:key>
  <xs:key name="keyCsapat">
    <xs:selector xpath="csapatok/csapat"/>
    <xs:field xpath="@cskod"/>
  </xs:key>

  <xs:keyref name="keyrefCS-J1" refer="keyJatekos">
    <xs:selector xpath="cs-js/cs-j"/>
    <xs:field xpath="@jkod"/>
  </xs:keyref>
  <xs:keyref name="keyrefCS-J2" refer="keyCsapat">
    <xs:selector xpath="cs-js/cs-j"/>
    <xs:field xpath="@cskod"/>
  </xs:keyref>

  <!-- Donto-Resztvesz(Merkozes) megszoritas-->
  <xs:key name="keyDonto">
    <xs:selector xpath="donto"/>
    <xs:field xpath="@dkod"/>
  </xs:key>
  <xs:key name="keyResztvesz">
    <xs:selector xpath="resztvesz"/>
    <xs:field xpath="@rkod"/>
  </xs:key>

  <xs:keyref name="keyrefResztvesz1" refer="keyMerkozes">
    <xs:selector xpath="resztvesz"/>
    <xs:field xpath="@mkod"/>
  </xs:keyref>
</xs:schema>
```

```

<xs:keyref name="keyrefResztvesz2" refer="keyDonto">
  <xs:selector xpath="resztvesz"/>
  <xs:field xpath="@dkod"/>
</xs:keyref>

<!-- Golok megszoritas -->
<xs:keyref name="keyrefR-CS1" refer="keyResztvesz">
  <xs:selector xpath="r-css/r-cs"/>
  <xs:field xpath="@rkod"/>
</xs:keyref>
<xs:keyref name="keyrefR-CS2" refer="keyCsapat">
  <xs:selector xpath="r-css/r-cs"/>
  <xs:field xpath="@cskod"/>
</xs:keyref>

</xs:element>

<!-- merkozes tipus -->
<xs:complexType name="merkozesTipus">
  <xs:sequence>
    <xs:element name="idopont" type="xs:date"/>
    <xs:element name="helyszin" type="helyszinTipus"/>
    <xs:element name="szabalytalansagok" type="szabalytalansagokTipus"/>
  </xs:sequence>
  <xs:attribute name="mkod" type="xs:string" use="required"/>
</xs:complexType>

<!-- helyszinTipus -->
<xs:complexType name="helyszinTipus">
  <xs:sequence>
    <xs:element name="ország" type="xs:string"/>
    <xs:element name="varos" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

<!-- szabalytalansag tipus -->
<xs:complexType name="szabalytalansagokTipus">
  <xs:sequence>
    <xs:element name="szabalytalansag" type="szabalytalansagTipus" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="szabalytalansagTipus">
  <xs:sequence>
    <xs:element name="idopont" type="xs:int"/>
    <xs:element name="lap" type="lapTipus"/>
  </xs:sequence>
  <xs:attribute name="mkod" type="xs:string" use="required"/>
</xs:complexType>

  <!-- lap tipus -->
  <xs:simpleType name="lapTipus">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Sarga"/>
      <xs:enumeration value="Piros"/>
    </xs:restriction>
  </xs:simpleType>

<!-- jatekvezetok-merkozes kapcsolat -->
<xs:complexType name="jatekvezetokTipus">
  <xs:sequence>
    <xs:element name="jatekvezeto" type="jatekvezetoTipus" minOccurs="1" maxOccurs="5"/>
  </xs:sequence>
</xs:complexType>

<!-- jatekvezeto tipus -->
<xs:complexType name="jatekvezetoTipus">
  <xs:sequence>
    <xs:element name="nev" type="xs:string"/>
    <xs:element name="ország" type="xs:string"/>
  </xs:sequence>
  <xs:attribute name="jkod" type="xs:string" use="required"/>
</xs:complexType>

```

```

<xs:complexType name="birokTipus">
  <xs:sequence>
    <xs:element name="biro" type="biroTipus" minOccurs="1" maxOccurs="5"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="biroTipus">
  <xs:sequence>
    <xs:element name="birotipus" type="birotipusTipus"/>
  </xs:sequence>
  <xs:attribute name="mkod" type="xs:string" use="required"/>
  <xs:attribute name="jkod" type="xs:string" use="required"/>
</xs:complexType>

<xs:simpleType name="birotipusTipus">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Jatekvezeto"/>
    <xs:enumeration value="Partjelzo"/>
    <xs:enumeration value="Videobiro"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="jatekosokTipus">
  <xs:sequence>
    <xs:element name="jatekos" type="jatekosTipus" minOccurs="2" maxOccurs="22"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="jatekosTipus">
  <xs:sequence>
    <xs:element name="nev" type="xs:string"/>
    <xs:element name="orszag" type="xs:string"/>
    <xs:element name="pozicio" type="pozicioTipus"/>
    <xs:element name="jatekosigazolvany">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:minLength value="1"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="jkod" type="xs:string" use="required"/>
</xs:complexType>

<xs:simpleType name="pozicioTipus">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Tamado"/>
    <xs:enumeration value="Kozeppanyas"/>
    <xs:enumeration value="Kapus"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="csapatokTipus">
  <xs:sequence>
    <xs:element name="csapat" type="csapatTipus" minOccurs="2" maxOccurs="2"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="csapatTipus">
  <xs:sequence>
    <xs:element name="nev" type="xs:string"/>
    <xs:element name="orszag" type="xs:string"/>
  </xs:sequence>
  <xs:attribute name="cskod" type="xs:string" use="required"/>
</xs:complexType>

<xs:complexType name="cs-jsTipus">
  <xs:sequence>
    <xs:element name="cs-j" type="cs-jTipus" minOccurs="2" maxOccurs="22"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="cs-jTipus">

```



```

        <xs:attribute name="jkod" type="xs:string" use="required"/>
        <xs:attribute name="cskod" type="xs:string" use="required"/>
    </xs:complexType>

    <xs:complexType name="dontoTipus">
        <xs:sequence>
            <xs:element name="ország" type="xs:string"/>
            <xs:element name="stadion" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="dkod" type="xs:string" use="required"/>
    </xs:complexType>

    <xs:complexType name="resztveszTipus">
        <xs:sequence>
            <xs:element name="nezoszam" type="xs:int" minOccurs="0" maxOccurs="1"/>
        </xs:sequence>
        <xs:attribute name="rkod" type="xs:string" use="required"/>
        <xs:attribute name="mkod" type="xs:string" use="required"/>
        <xs:attribute name="dkod" type="xs:string" use="required"/>
    </xs:complexType>

    <xs:complexType name="r-cssTipus">
        <xs:sequence>
            <xs:element name="r-cs" type="r-csTipus" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="r-csTipus">
        <xs:sequence>
            <xs:element name="gol" type="xs:int"/>
        </xs:sequence>
        <xs:attribute name="rkod" type="xs:string" use="required"/>
        <xs:attribute name="cskod" type="xs:string" use="required"/>
    </xs:complexType>
</xs:schema>

```

DOMReader kódja, valamint konzol eredménye:

Fájlnev: DomReadL3N7UO.java

Kilistázza a mérkőzésen lévő bírót, a csapatokat, a játékosokat és ezek adatait valamint a döntő helyét és nézőszámát ha van.

```

package hu.domparse.l3n7uo;

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 DomReadL3N7UO {
    public static void main(String argv[]) throws ParserConfigurationException, SAXException,
    IOException {

        //XML Fájl meghívása.
        File xmlFile = new File("XMLL3N7UO.xml");

        //Dokument builder.
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = factory.newDocumentBuilder();

        //XML Fájl betöltése a dBuilderbe.

```

```

Document doc = dBuilder.parse(xmlFile);
doc.getDocumentElement().normalize();

System.out.println("Root element: " + doc.getDocumentElement().getNodeName());

//jatekvezetok kilistázása.
System.out.println("\n-----Jatekvezetok-----");

NodeList jatekvezetoList = doc.getElementsByTagName("jatekvezeto");
for(int i=0; i<jatekvezetoList.getLength(); i++) {
    Node nNode = jatekvezetoList.item(i);
    printJatekvezeto(nNode);
}

//csapatok kilistázása.
System.out.println("\n-----Csapatok-----");

NodeList csapatList = doc.getElementsByTagName("csapat");
for(int i=0; i<csapatList.getLength(); i++) {
    Node nNode = csapatList.item(i);
    printCsapat(nNode);
}

//jatekosok kilistázása.
System.out.println("\n-----Jatekosok-----");

NodeList jatekosList = doc.getElementsByTagName("jatekos");
for(int i=0; i<jatekosList.getLength(); i++) {
    Node nNode = jatekosList.item(i);
    printJatekos(nNode);
}

//donto kilistázása.
System.out.println("\n-----Donto-----");

NodeList dontoList = doc.getElementsByTagName("donto");
for(int i=0; i<dontoList.getLength(); i++) {
    Node nNode = dontoList.item(i);
    String dkod = printDonto(nNode);

    NodeList resztveszList = doc.getElementsByTagName("resztvesz");
    for(int j=0; j<resztveszList.getLength(); j++) {
        Node nNode2 = resztveszList.item(j);
        printResztvesz(nNode2, dkod);
    }
}

}

private static void printJatekvezeto(Node nNode) {
    if(nNode.getNodeType() == Node.ELEMENT_NODE) {
        Element elem = (Element) nNode;
        String jkod = elem.getAttribute("jkod");

        Node nNode1 = elem.getElementsByTagName("nev").item(0);
        String nev = nNode1.getTextContent();

        Node nNode2 = elem.getElementsByTagName("ország").item(0);
        String orszag = nNode2.getTextContent();

        System.out.printf("JKOD: %s\n", jkod);
        System.out.printf("Nev: %s\n", nev);
        System.out.printf("Orszag: %s\n", orszag);
        System.out.println("");
    }
}

private static void printCsapat(Node nNode) {
    if(nNode.getNodeType() == Node.ELEMENT_NODE) {
        Element elem = (Element) nNode;
        String cskod = elem.getAttribute("cskod");
    }
}

```

```

        Node nNode1 = elem.getElementsByTagName("nev").item(0);
        String nev = nNode1.getTextContent();

        Node nNode2 = elem.getElementsByTagName("orszag").item(0);
        String orszag = nNode2.getTextContent();

        System.out.printf("CSKOD: %s%n", cskod);
        System.out.printf("Nev: %s%n", nev);
        System.out.printf("Orszag: %s%n", orszag);
        System.out.println("");
    }
}

private static void printJatekos(Node nNode) {
    if(nNode.getNodeType() == Node.ELEMENT_NODE) {
        Element elem = (Element) nNode;
        String jkod = elem.getAttribute("jkod");

        Node nNode1 = elem.getElementsByTagName("nev").item(0);
        String nev = nNode1.getTextContent();

        Node nNode2 = elem.getElementsByTagName("orszag").item(0);
        String orszag = nNode2.getTextContent();

        Node nNode3 = elem.getElementsByTagName("pozicio").item(0);
        String pozicio = nNode3.getTextContent();

        Node nNode4 = elem.getElementsByTagName("jatekosigazolvany").item(0);
        String jatekosigazolvany = nNode4.getTextContent();

        System.out.printf("JKOD: %s%n", jkod);
        System.out.printf("Nev: %s%n", nev);
        System.out.printf("Orszag: %s%n", orszag);
        System.out.printf("Pozicio: %s%n", pozicio);
        System.out.printf("Jatekosigazolvany: %s%n", jatekosigazolvany);
        System.out.println("");
    }
}

private static String printDonto(Node nNode) {
    String dkod = null;

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

        Node nNode1 = elem.getElementsByTagName("orszag").item(0);
        String orszag = nNode1.getTextContent();

        Node nNode2 = elem.getElementsByTagName("stadion").item(0);
        String stadion = nNode2.getTextContent();

        System.out.printf("DKOD: %s%n", dkod);
        System.out.printf("Orszag: %s%n", orszag);
        System.out.printf("Stadion: %s%n", stadion);
    }

    return dkod;
}

private static void printResztvesz(Node nNode, String dkod) {
    if(nNode.getNodeType() == Node.ELEMENT_NODE) {
        Element elem = (Element) nNode;
        String dkod2 = elem.getAttribute("dkod");

        if(dkod2.equals(dkod)) {
            Node nNode2 = elem.getElementsByTagName("nezoszam").item(0);
            String nezoszam = nNode2.getTextContent();

            System.out.printf("Nezoszam: %s%n", nezoszam);
            System.out.println("");
        }
    }
}

```

```
}  
}
```

<terminated> DomReadL3N7UO (1) [Java Application] C:\Users\horan\jdk-13.0.2\bin\javaw.exe (2021. nov. 28. 12:30:20 – 12:30:21)

JKOD: J02
Nev: Toni Kroos
Ország: Németország
Pozíció: Középpályás
Jatekosigazolvány: ABC456

JKOD: J03
Nev: Thibaut Courtois
Ország: Belgium
Pozíció: Kapus
Jatekosigazolvány: ABC789

JKOD: J04
Nev: Robert Lewandowski
Ország: Lengyelország
Pozíció: Támadó
Jatekosigazolvány: CBA123

JKOD: J05
Nev: Leon Goretzka
Ország: Németország
Pozíció: Középpályás
Jatekosigazolvány: CBA456

JKOD: J06
Nev: Manuel Neuer
Ország: Németország
Pozíció: Kapus
Jatekosigazolvány: CBA789

DomReadL3N7UO.java × DomModifyL3N7UO.java × XMLL3N7UO.xml × XMLL3N7UO.xml × DomQueryL3N7UO.java

18

Problems Javadoc Declaration Console × Properties

<terminated> DomReadL3N7UO (1) [Java Application] C:\Users\horan\jdk-13.0.2\bin\javaw.exe (2021. nov. 28. 12:30:20 – 12:30:21)

Root element: document

-----Jatekvezetok-----

JKOD: B01
Nev: Smith Jhon
Ország: Anglia

JKOD: B02
Nev: Dany Lewis
Ország: Skócia

JKOD: B03
Nev: Big Pete
Ország: Anglia

-----Csapatok-----

CSKOD: CS01
Nev: Real Madrid
Ország: Spanyolország

CSKOD: CS02
Nev: Bayern München
Ország: Németország

-----Jatekosok-----

JKOD: J01
Nev: Karim Benzema
Ország: Franciaország
Pozíció: Támadó
Jatekosigazolvány: ABC123

JKOD: J02

-----Donto-----

DKOD: D01
Ország: Oroszország
Stadion: Saint Petersburg Stadium
Nezozam: 10000000

DOMModify kódja, valamint konzol eredménye:

Fájlnev: DomModifyL3N7UO.java

Kilistázza a jelenlegi játékosokat és adataikat majd további játékosokat viszünk fel hozzájuk. Ezzel megtörténik az XML módosítása majd a módosított adatokat elmentjük egy új XML fájlba.

```
package hu.domparsed.l3n7uo;

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.OutputKeys;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerConfigurationException;
import javax.xml.transform.TransformerException;
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.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;

public class DomModifyL3N7UO {

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

        //XML Fájl meghívása.
        File xmlFile = new File("XMLL3N7UO.xml");

        //Dokument builder.
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = factory.newDocumentBuilder();

        //XML fájl betöltése a dBuilderbe.
        Document doc = dBuilder.parse(xmlFile);
        doc.getDocumentElement().normalize();

        //Módosítás előtt:
        //Minden játékosok kilistázása.
        System.out.println("\n-----Jatekosok-----");

        NodeList jatekosList = doc.getElementsByTagName("jatekos");
        for(int i=0; i<jatekosList.getLength(); i++) {
            Node nNode = jatekosList.item(i);
            printJatekos(nNode);
        }

        //Módosítás után:
        //Minden játékosok-hoz feltöltés majd kilistázása.

        Node jNode = doc.getElementsByTagName("jatekosok").item(0);
        Element jatekosok = (Element) jNode;

        Node addJatekos1 = createJatekos(doc, "J07", "Eden Hazard", "Belgium", "Tamado", "ABC010");
        jatekosok.appendChild(addJatekos1);

        Node addJatekos2 = createJatekos(doc, "J08", "Thomas Müller", "Németország", "Kozepallyas",
            "CBA010");
        jatekosok.appendChild(addJatekos2);

        Node addJatekos3 = createJatekos(doc, "J09", "Luka Modric", "Horvátország", "Kozepallyas",
            "ABC020");
        jatekosok.appendChild(addJatekos3);
    }
}
```

```

Node addJatekos4 = createJatekos(doc, "J10", "Leroy Sané", "Németország", "Tamado", "CBA020");
jatekosok.appendChild(addJatekos4);

System.out.println("\n-----Jatekosok-----");

jatekosList = doc.getElementsByTagName("jatekos");
for(int i=0; i<jatekosList.getLength(); i++) {
    Node nNode = jatekosList.item(i);
    printJatekos(nNode);
}

saveAsDoc(doc, "XMLL3N7UO_Modify.xml");
}

private static void printJatekos(Node nNode) {
    if(nNode.getNodeType() == Node.ELEMENT_NODE) {
        Element elem = (Element) nNode;
        String jkod = elem.getAttribute("jkod");

        Node nNode1 = elem.getElementsByTagName("nev").item(0);
        String nev = nNode1.getTextContent();

        Node nNode2 = elem.getElementsByTagName("ország").item(0);
        String orszag = nNode2.getTextContent();

        Node nNode3 = elem.getElementsByTagName("pozicio").item(0);
        String pozicio = nNode3.getTextContent();

        Node nNode4 = elem.getElementsByTagName("jatekosigazolvany").item(0);
        String jatekosigazolvany = nNode4.getTextContent();

        System.out.printf("JKOD: %s\n", jkod);
        System.out.printf("Nev: %s\n", nev);
        System.out.printf("Orszag: %s\n", orszag);
        System.out.printf("Pozicio: %s\n", pozicio);
        System.out.printf("Jatekosigazolvany: %s\n", jatekosigazolvany);
        System.out.println("");
    }
}

private static void printJatekos(Node nNode, String pozicio) {
    if(nNode.getNodeType() == Node.ELEMENT_NODE) {
        Element elem = (Element) nNode;
        String jkod = elem.getAttribute("jkod");

        Node nNode1 = elem.getElementsByTagName("nev").item(0);
        String nev = nNode1.getTextContent();

        Node nNode2 = elem.getElementsByTagName("ország").item(0);
        String orszag = nNode2.getTextContent();

        Node nNode3 = elem.getElementsByTagName("pozicio").item(0);
        String pozicio2 = nNode3.getTextContent();

        Node nNode4 = elem.getElementsByTagName("jatekosigazolvany").item(0);
        String jatekosigazolvany = nNode4.getTextContent();

        if(pozicio2.equals(pozicio)) {
            System.out.printf("JKOD: %s\n", jkod);
            System.out.printf("Nev: %s\n", nev);
            System.out.printf("Orszag: %s\n", orszag);
            System.out.printf("Pozicio: %s\n", pozicio);
            System.out.printf("Jatekosigazolvany: %s\n", jatekosigazolvany);
            System.out.println("");
        }
    }
}

private static Node createElement(Document doc, String name, String value) {
    Element node = doc.createElement(name);
    node.appendChild(doc.createTextNode(value));

    return node;
}

```

```

private static Node createJatekos(Document doc, String jkod, String nev, String orszag, String
pozicio, String jatekosigazolvany) {
    Element jatekos = doc.createElement("jatekos");
    jatekos.setAttribute("jkod", jkod);
    jatekos.appendChild(createElement(doc, "nev", nev));
    jatekos.appendChild(createElement(doc, "orszag", orszag));
    jatekos.appendChild(createElement(doc, "pozicio", pozicio));
    jatekos.appendChild(createElement(doc, "jatekosigazolvany", jatekosigazolvany));

    return jatekos;
}

private static void saveAsDoc(Document doc, String filename) throws TransformerException {
    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");

    DOMSource source = new DOMSource(doc);

    File xmlFile = new File(filename);

    StreamResult console = new StreamResult(System.out);
    StreamResult file = new StreamResult(xmlFile);

    transf.transform(source, console);
    transf.transform(source, file);
}
}

```

The screenshot shows an IDE with several tabs: DomReadL3N7UO.java, DomModifyL3N7UO.java, XMLL3N7UO.xml, XMLL3N7UO.xml, and DomQueryL3N7UO.java. The main window displays the XML output of the application, which is a list of players (Jatekosok) with their attributes (JKOD, Nev, Orszag, Pozicio, Jatekosigazolvany). The console window shows the XML output, and the Properties window shows the XML metadata.

```

System.out.println("\n-----Jatekosok-----");

<terminated> DomModifyL3N7UO [Java Application] C:\Users\horan\jdk-13.0.2\bin\javaw.exe (2021. nov. 28. 12:42:27 - 12:42:28)

JKOD: J06
Nev: Manuel Neuer
Orszag: Németország
Pozicio: Kapus
Jatekosigazolvany: CBA789

JKOD: J07
Nev: Eden Hazard
Orszag: Belgium
Pozicio: Tamado
Jatekosigazolvany: ABC010

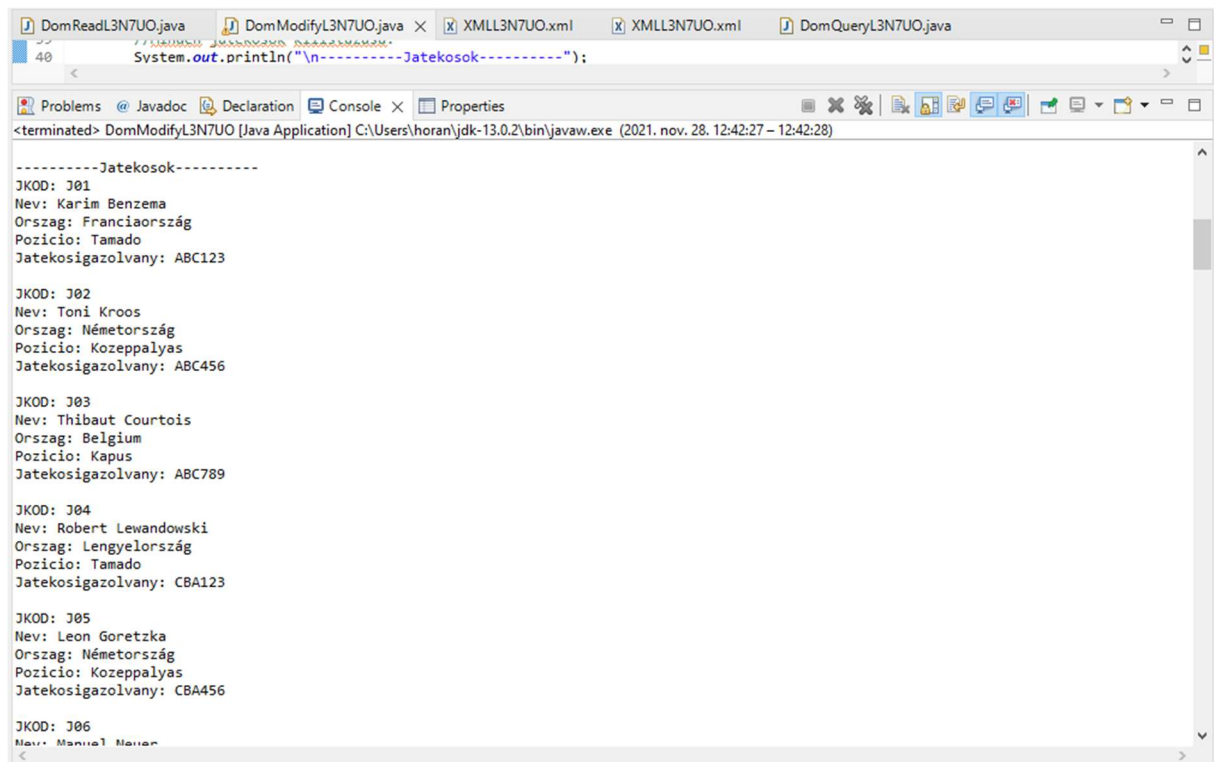
JKOD: J08
Nev: Thomas Müller
Orszag: Németország
Pozicio: Kozepalyas
Jatekosigazolvany: CBA010

JKOD: J09
Nev: Luka Modric
Orszag: Horvátország
Pozicio: Kozepalyas
Jatekosigazolvany: ABC020

JKOD: J10
Nev: Leroy Sané
Orszag: Németország
Pozicio: Tamado
Jatekosigazolvany: CBA020

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="XMLSchemaL3N7UO.xsd">

```



```
System.out.println("\n-----Jatekosok-----");

-----Jatekosok-----
JKOD: J01
Nev: Karim Benzema
Orszag: Franciaország
Pozicio: Tamado
Jatekosigazolvany: ABC123

JKOD: J02
Nev: Toni Kroos
Orszag: Németország
Pozicio: Kozepalyas
Jatekosigazolvany: ABC456

JKOD: J03
Nev: Thibaut Courtois
Orszag: Belgium
Pozicio: Kapus
Jatekosigazolvany: ABC789

JKOD: J04
Nev: Robert Lewandowski
Orszag: Lengyelország
Pozicio: Tamado
Jatekosigazolvany: CBA123

JKOD: J05
Nev: Leon Goretzka
Orszag: Németország
Pozicio: Kozepalyas
Jatekosigazolvany: CBA456

JKOD: J06
Nev: Manuel Neuer
```

DOMQuery kódja, valamint a konzol eredménye:

Fájlnév: DomQueryL3N7UO.java

Kilistázza az összes játékost majd pozíciójukra szűrve, elsőnek a támadókat majd a középpályásokat és végül a kapusokat írja ki.

```
package hu.domparse.l3n7uo;

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 DomQueryL3N7UO {
    public static void main(String argv[]) throws ParserConfigurationException, SAXException,
        IOException {

        //XML Fájl meghívása.
        File xmlFile = new File("XMLL3N7UO.xml");

        //Dokument builder.
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = factory.newDocumentBuilder();

        //XML fájl betöltése a dBuilderbe.
        Document doc = dBuilder.parse(xmlFile);
        doc.getDocumentElement().normalize();

        //Minden jatekosok kilistázása.
        System.out.println("\n-----Jatekosok-----");

        NodeList jatekosList = doc.getElementsByTagName("jatekos");
```



```

for(int i=0; i<jatekosList.getLength(); i++) {
    Node nNode = jatekosList.item(i);
    printJatekos(nNode);
}

//Szűrések.
//Minden "tamado" jatekosok kilistázása.
System.out.println("\n-----Jatekosok szures utan:-----\n");
System.out.println("\n-----Tamado Jatekosok-----");

NodeList jatekosTamadoList = doc.getElementsByTagName("jatekos");
for(int i=0; i<jatekosTamadoList.getLength(); i++) {
    Node nNode = jatekosTamadoList.item(i);
    printJatekos(nNode, "Tamado");
}

//Minden "kozeppalyas" jatekosok kilistázása.
System.out.println("\n-----Kozeppalyas Jatekosok-----");

NodeList jatekosKozeppalyasList = doc.getElementsByTagName("jatekos");
for(int i=0; i<jatekosKozeppalyasList.getLength(); i++) {
    Node nNode = jatekosKozeppalyasList.item(i);
    printJatekos(nNode, "Kozeppalyas");
}

//Minden "kapus" jatekosok kilistázása.
System.out.println("\n-----Kapus Jatekosok-----");

NodeList jatekosKapusList = doc.getElementsByTagName("jatekos");
for(int i=0; i<jatekosKapusList.getLength(); i++) {
    Node nNode = jatekosKapusList.item(i);
    printJatekos(nNode, "Kapus");
}
}

private static void printJatekos(Node nNode) {
    if(nNode.getNodeType() == Node.ELEMENT_NODE) {
        Element elem = (Element) nNode;
        String jkod = elem.getAttribute("jkod");

        Node nNode1 = elem.getElementsByTagName("nev").item(0);
        String nev = nNode1.getTextContent();

        Node nNode2 = elem.getElementsByTagName("ország").item(0);
        String orszag = nNode2.getTextContent();

        Node nNode3 = elem.getElementsByTagName("pozicio").item(0);
        String pozicio = nNode3.getTextContent();

        Node nNode4 = elem.getElementsByTagName("jatekosigazolvany").item(0);
        String jatekosigazolvany = nNode4.getTextContent();

        System.out.printf("JKOD: %s\n", jkod);
        System.out.printf("Nev: %s\n", nev);
        System.out.printf("Orszag: %s\n", orszag);
        System.out.printf("Pozicio: %s\n", pozicio);
        System.out.printf("Jatekosigazolvany: %s\n", jatekosigazolvany);
        System.out.println("");
    }
}

private static void printJatekos(Node nNode, String pozicio) {
    if(nNode.getNodeType() == Node.ELEMENT_NODE) {
        Element elem = (Element) nNode;
        String jkod = elem.getAttribute("jkod");

        Node nNode1 = elem.getElementsByTagName("nev").item(0);
        String nev = nNode1.getTextContent();

        Node nNode2 = elem.getElementsByTagName("ország").item(0);
        String orszag = nNode2.getTextContent();

        Node nNode3 = elem.getElementsByTagName("pozicio").item(0);

```

```

String pozicio2 = nNode3.getTextContent();

Node nNode4 = elem.getElementsByTagName("jatekosigazolvany").item(0);
String jatekosigazolvany = nNode4.getTextContent();

if(pozicio2.equals(pozicio)) {
    System.out.printf("JKOD: %s\n", jkod);
    System.out.printf("Nev: %s\n", nev);
    System.out.printf("Orszag: %s\n", orszag);
    System.out.printf("Pozicio: %s\n", pozicio);
    System.out.printf("Jatekosigazolvany: %s\n", jatekosigazolvany);
    System.out.println("");
}
}
}
}
}

```

The screenshot shows an IDE with several tabs: DomReadL3N7UO.java, DomModifyL3N7UO.java, XMLL3N7UO.xml, XMLL3N7UO.xml, and DomQueryL3N7UO.java. The active tab is DomQueryL3N7UO.java, which contains the following code:

```

package hu.domparsel3n7uo;

```

The console window shows the output of the program, which is a list of football players and their statistics, grouped by position. The output is as follows:

```

<terminated> DomQueryL3N7UO [Java Application] C:\Users\horan\jdk-13.0.2\bin\javaw.exe (2021. nov. 28. 12:48:16 – 12:48:16)

|
|-----Jatekosok-----|
JKOD: J01
Nev: Karim Benzema
Orszag: Franciaország
Pozicio: Tamado
Jatekosigazolvany: ABC123

JKOD: J02
Nev: Toni Kroos
Orszag: Németország
Pozicio: Kozepalyas
Jatekosigazolvany: ABC456

JKOD: J03
Nev: Thibaut Courtois
Orszag: Belgium
Pozicio: Kapus
Jatekosigazolvany: ABC789

JKOD: J04
Nev: Robert Lewandowski
Orszag: Lengyelország
Pozicio: Tamado
Jatekosigazolvany: CBA123

JKOD: J05
Nev: Leon Goretzka
Orszag: Németország
Pozicio: Kozepalyas
Jatekosigazolvany: CBA456

JKOD: J06
Nev: Manuel Neuer
Orszag: Németország
Pozicio: Kapus
Jatekosigazolvany: CBA789

|-----Jatekosok szures utan:-----|

|-----Tamado Jatekosok-----|
JKOD: J01
Nev: Karim Benzema
Orszag: Franciaország
Pozicio: Tamado
Jatekosigazolvany: ABC123

JKOD: J04
Nev: Robert Lewandowski
Orszag: Lengyelország
Pozicio: Tamado
Jatekosigazolvany: CBA123

|-----Kozepalyas Jatekosok-----|
JKOD: J02
Nev: Toni Kroos
Orszag: Németország
Pozicio: Kozepalyas
Jatekosigazolvany: ABC456

JKOD: J05
Nev: Leon Goretzka

```

<terminated> DomQueryL3N7UO [Java Application] C:\Users\horan\jdk-13.0.2\bin\javaw.exe (2021. nov. 28. 12:48:16 – 12:48:16)

Nev: Róbert Lewandowski
Ország: Lengyelország
Pozíció: Tamado
Jatekosigazolvany: CBA123

-----Kozepallyas Jatekosok-----

JKOD: J02
Nev: Toni Kroos
Ország: Németország
Pozíció: Kozepallyas
Jatekosigazolvany: ABC456

JKOD: J05
Nev: Leon Goretzka
Ország: Németország
Pozíció: Kozepallyas
Jatekosigazolvany: CBA456

-----Kapus Jatekosok-----

JKOD: J03
Nev: Thibaut Courtois
Ország: Belgium
Pozíció: Kapus
Jatekosigazolvany: ABC789

JKOD: J06
Nev: Manuel Neuer
Ország: Németország
Pozíció: Kapus
Jatekosigazolvany: CBA789