

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Labdarúgó Világbajnokság nyilvántartása

Készítette: Szkárosi Szilárd

Neptun kód: DLWGQZ

Gyakorlatvezető: Dr. Bednarik László

Gyakorlat időpontja: Szerda 10:00-12:00

Képzés: Mérnökinformatikus BSc nappali

Tartalomjegyzék

1. Feladat leírása	3
1a.) Az egyedek tulajdonságai.....	3
1b.) A feladat ER Modellje	5
1c.) Az egyedek közötti kapcsolatok.....	5
2. Az ER modell konvertálása XDM modellé.....	6
3. XML dokumentum készítése.....	6
4. XMLSchema készítése XML dokumentum alapján	6
5. DOM program készítése JAVA környezetben.....	18
5a.) DOM adatolvasás	18
5b.) DOM adatlekérdezés	34
5c.) DOM adatmódosítás.....	41
5d.) DOM adatírás	49

1. Feladat leírása

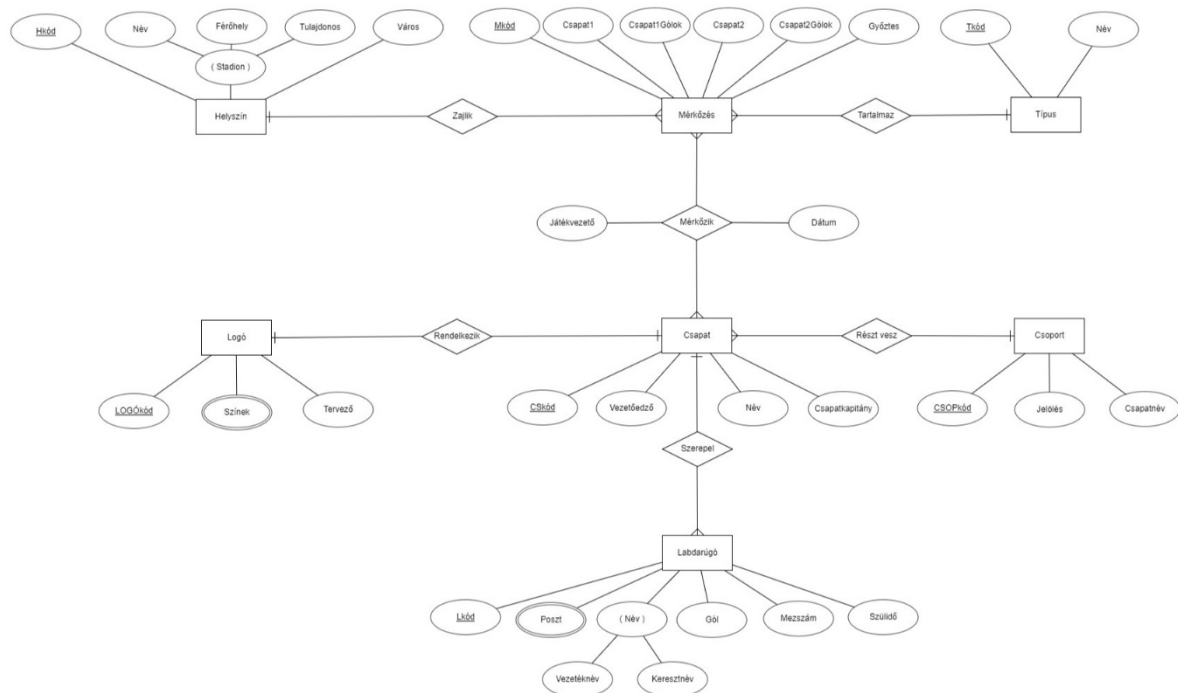
A féléves beadandó tematikája egy Labdarúgó Világbajnokság csoportjainak, csapatainak és labdarúgóinak nyilvántartása XML-ben. Az XML dokumentumban tudunk beolvasni, írni, lekérdezni és módosítani.

1a.) Az egyedek tulajdonságai

- **Csapat:**
 - **CSkód:** A Csapat egyed elsődleges kulcsa.
 - **Név:** A csapat neve.
 - **Vezetőedző:** Az edző neve.
 - **Csapatkapitány:** A csapatkapitány neve. Minden csapat rendelkezik egy csapatkapitánnyal.
- **Logó:**
 - **LOGÓkód:** A Logó egyed elsődleges kulcsa.
 - **Színek:** Többértékű tulajdonság, amely azt jelenti, hogy a logóban többféle színt tartalmaz. *Például a francia válogatott zászlaján 3 darab különböző szín található.*
 - **Tervező:** A tervező neve.
- **Csoport:**
 - **CSOPkód:** A Csoport egyed elsődleges kulcsa.
 - **Csapatnév:** A csapat neve.
 - **Jelölés:** A csoport nevét tárolja el. A csapatok sorsolás folyamán bekerülnek egy adott csoportba. A csoportok A-tól H-ig vannak. *Például a 2022-es VB-n a Portugál válogatott a H-csoportban szerepel.*
- **Labdarúgó:**
 - **Lkód:** A Labdarúgó egyed elsődleges kulcsa.
 - **Poszt:** Többértékű tulajdonság, amely azt jelenti, hogy a labdarúgó több poszton is játszhat. *Például Camavinga eredetileg középpályás, de tud balszélső védőként is játszani.* A labdarúgó az alábbi poszton játszhat: kapus, védő, középpályás és csatár.
 - **Név:** Összetett tulajdonság, mivel a labdarúgónak van vezeték- és keresztnéve.
 - **Gól:** Az adott labdarúgó hány gólt szerzett a VB-n.
 - **Mezszám:** A labdarúgó mezszáma.

- **Szülidő:** A labdarúgó születési ideje.
- **Mérkőzés:**
 - **Mkód:** A Mérkőzés egyed elsődleges kulcsa.
 - **Csapat1:** Az első csapat neve.
 - **Csapat1Gólok:** Az első csapat meccsen szerzett góljainak száma
 - **Csapat2:** A második csapat neve.
 - **Csapat2Gólok:** Az első csapat meccsen szerzett góljainak száma
 - **Győztes:** A győztes csapat neve.
- **Helyszín:**
 - **Hkód:** A Helyszín egyed elsődleges kulcsa.
 - **Stadion:** Összetett tulajdonság, amelynek van neve, férőhelye és tulajdonosa.
 - **Város:** A város neve.
- **Típus:**
 - **Tkód:** A Típus egyed elsődleges kulcsa.
 - **Név:** A mérkőzés típusának a neve. Lehet csoport-, nyolcad-, negyed-, elő- és döntő.
- **Mérkőzők:**
 - **Dátum:** Mérkőzés dátuma.
 - **Játékvezető:** Játékvezető neve.

1b.) A feladat ER Modellje



1c.) Az egyedek közötti kapcsolatok

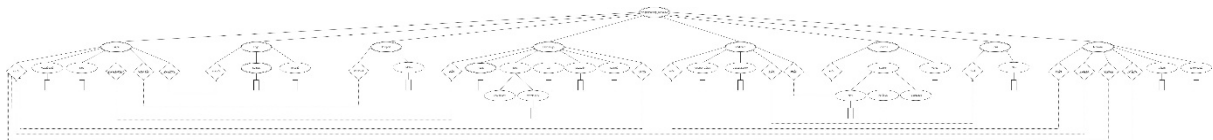
- **Csapat és Csoport:**
 - A **Csapat** és a **Csoport** között **1:N** típusú kapcsolat van, mivel egy csoportban több csapat is szerepelhet, de egy csapat csak egy csoportban játszhat. A VB-n egy csoportban összesen 4 darab csapat szokott lenni.
- **Csapat és Labdarúgó:**
 - A **Csapat** és a **Labdarúgó** között **1:N** típusú kapcsolat van, mivel egy csapatban több labdarúgó is szerepelhet, de egy labdarúgó csak egy csapatban játszhat. *Például Luka Modric csak a horvát válogatott csapatában játszhat (mivel horvát születésű), de más csapatban nem.*
- **Csapat és Logó:**
 - A **Csapat** és a **Logó** között **1:1** típusú kapcsolat van, mivel egy csapatnak egy logója lehet, és egy logóhoz egy csapat tartozhat.
- **Csapat és Mérkőzés:**
 - A **Csapat** és a **Mérkőzés** között **N:N** típusú kapcsolat van, mivel több csapat szokott játszani több mérkőzést.
- **Mérkőzés és Helyszín:**

- A **Mérkőzés** és a **Helyszín** között **1:N** típusú kapcsolat van, mivel egy mérkőzés egy helyszínen szokott zajlani, de egy helyszínen több mérkőzést is játszhatnak.
- **Mérkőzés és Típus:**
 - A **Mérkőzés** és a **Típus** között **1:N** típusú kapcsolat van, mivel egy mérkőzésnek egy típusa van, de egy típus több mérkőzéshez tartozhat.

2. Az ER modell konvertálása XDM modellé

XDM modellnél háromféle jelölés használunk: ellipszist, rombuszt és téglalapot. Az ellipszis jelöli az elemeket, minden egyedből elem lesz, illetve a tulajdonságokból is. A rombusz jelöli az attribútumokat, amelyek a kulcs tulajdonságokból keletkeznek. A téglalap jelöli a szöveget, amely majd az XML dokumentumban fog megjelenni. Azoknak az elemeknek, amelyek többször is előfordulhatnak, a jelölése dupla ellipszissel történik. Az idegenkulcsok és a kulcsok közötti kapcsolatot szaggatott vonalas nyíllal jelöljük.

XDM modell:



3. XML dokumentum készítése

Az XDM modell alapján készítettem el az XML dokumentumot. Legelőször a gyökérelemmel kezdtem, amelynek a „Világ bajnokság_DLWGQZ” nevet adtam. Ezek után a gyermekelemeiből eltérő módon hoztam létre példányokat.

XML dokumentum forráskódja:

```
<?xml version="1.0" encoding="UTF-8"?>
<VilágBajnokság_DLWGQZ xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
xs:noNamespaceSchemaLocation="XMLSchemaDLWGQZ.xsd">
  <!--Csapatok-->
  <Csapat CSkód="1" Csapatkapitány="CR7" CSOPkód="11" LOGÓkód="21">
    <Vezetőedző>Roberto Martínez</Vezetőedző>
    <Név>Portugália</Név>
  </Csapat>
  <Csapat CSkód="2" Csapatkapitány="TS1" CSOPkód="12" LOGÓkód="22">
    <Vezetőedző>Hansi Flick</Vezetőedző>
    <Név>Németország</Név>
  </Csapat>
  <Csapat CSkód="3" Csapatkapitány="KM10" CSOPkód="13" LOGÓkód="23">
    <Vezetőedző>Didier Deschamp</Vezetőedző>
    <Név>Franciaország</Név>
```

```
</Csapat>
<Csapat CSkód="4" Csapatkapitány="HK9" CSOPkód="14" LOGÓkód="24">
  <Vezetőedző>Gareth Southgate</Vezetőedző>
  <Név>Anglia</Név>
</Csapat>
<Csapat CSkód="5" Csapatkapitány="LM10" CSOPkód="15" LOGÓkód="25">
  <Vezetőedző>Lionel Scaloni</Vezetőedző>
  <Név>Argentína</Név>
</Csapat>
<!--Logók-->
<Logó LOGÓkód="21">
  <Színek>Piros</Színek>
  <Színek>Kék</Színek>
  <Színek>Zöld</Színek>
  <Tervező>António Modesto</Tervező>
</Logó>
<Logó LOGÓkód="22">
  <Színek>Fekete</Színek>
  <Színek>Piros</Színek>
  <Színek>Arany</Színek>
  <Tervező>Anton Stankowski</Tervező>
</Logó>
<Logó LOGÓkód="23">
  <Színek>Kék</Színek>
  <Színek>Fehér</Színek>
  <Színek>Piros</Színek>
  <Tervező>Raymond Savignac</Tervező>
</Logó>
<Logó LOGÓkód="24">
  <Színek>Fehér</Színek>
  <Színek>Kék</Színek>
  <Színek>Piros</Színek>
  <Tervező>William C. Gibbons</Tervező>
</Logó>
<Logó LOGÓkód="25">
  <Színek>Égkék</Színek>
  <Színek>Fehér</Színek>
  <Színek>Sárga</Színek>
  <Tervező>Salvador Dellutri</Tervező>
</Logó>
<!--Csoportok-->
<Csoport CSOPkód="11">
  <Jelölés>A</Jelölés>
</Csoport>
<Csoport CSOPkód="12">
  <Jelölés>B</Jelölés>
</Csoport>
<Csoport CSOPkód="13">
```

```
<Jelölés>C</Jelölés>
</Csoport>
<Csoport CSOPkód="14">
  <Jelölés>D</Jelölés>
</Csoport>
<Csoport CSOPkód="15">
  <Jelölés>E</Jelölés>
</Csoport>
<!--Labdarúgók-->
<!--PORTUGÁL Labdarúgók-->
<Labdarúgó Lkód="CR7" CSkód="1">
  <Poszt>Középcsatár</Poszt>
  <Név>
    <Keresztnév>Cristiano</Keresztnév>
    <Vezetéknév>Ronaldo</Vezetéknév>
  </Név>
  <Gól>10</Gól>
  <Mezszám>7</Mezszám>
  <Szülidő>1985.02.05</Szülidő>
</Labdarúgó>
<Labdarúgó Lkód="RN18" CSkód="1">
  <Poszt>Középpályás</Poszt>
  <Poszt>Védekező középpályás</Poszt>
  <Név>
    <Keresztnév>Ruben</Keresztnév>
    <Vezetéknév>Neves</Vezetéknév>
  </Név>
  <Gól>2</Gól>
  <Mezszám>18</Mezszám>
  <Szülidő>1997.03.13</Szülidő>
</Labdarúgó>
<!--NÉMET Labdarúgók-->
<Labdarúgó Lkód="TS1" CSkód="2">
  <Poszt>Kapus</Poszt>
  <Név>
    <Keresztnév>Marc-André</Keresztnév>
    <Vezetéknév>ter Stegen</Vezetéknév>
  </Név>
  <Gól>0</Gól>
  <Mezszám>22</Mezszám>
  <Szülidő>1992.04.05</Szülidő>
</Labdarúgó>
<Labdarúgó Lkód="JK6" CSkód="2">
  <Poszt>Középpályás</Poszt>
  <Poszt>Jobb szélső védő</Poszt>
  <Név>
    <Keresztnév>Joshua</Keresztnév>
    <Vezetéknév>Kimmich</Vezetéknév>
```



```

        </Név>
        <Gól>3</Gól>
        <Mezszám>6</Mezszám>
        <Szülidő>1995.02.08</Szülidő>
    </Labdarúgó>
    <!--FRANCIA Labdarúgók-->
    <Labdarúgó Lkód="KM10" CSkód="3">
        <Poszt>Középcsatár</Poszt>
        <Név>
            <Keresztnév>Kylian</Keresztnév>
            <Vezetéknév>Mbappé</Vezetéknév>
        </Név>
        <Gól>9</Gól>
        <Mezszám>10</Mezszám>
        <Szülidő>1998.12.20</Szülidő>
    </Labdarúgó>
    <Labdarúgó Lkód="EC25" CSkód="3">
        <Poszt>Középpályás</Poszt>
        <Poszt>Védekező középpályás</Poszt>
        <Poszt>Bal szélső védő</Poszt>
        <Név>
            <Keresztnév>Eduardo</Keresztnév>
            <Vezetéknév>Camavinga</Vezetéknév>
        </Név>
        <Gól>3</Gól>
        <Mezszám>25</Mezszám>
        <Szülidő>2002.11.10</Szülidő>
    </Labdarúgó>
    <!--ANGOL Labdarúgók-->
    <Labdarúgó Lkód="HK9" CSkód="4">
        <Poszt>Középcsatár</Poszt>
        <Név>
            <Keresztnév>Harry</Keresztnév>
            <Vezetéknév>Kane</Vezetéknév>
        </Név>
        <Gól>7</Gól>
        <Mezszám>9</Mezszám>
        <Szülidő>1993.07.28</Szülidő>
    </Labdarúgó>
    <Labdarúgó Lkód="JB10" CSkód="4">
        <Poszt>Középpályás</Poszt>
        <Név>
            <Keresztnév>Jude</Keresztnév>
            <Vezetéknév>Bellingham</Vezetéknév>
        </Név>
        <Gól>6</Gól>
        <Mezszám>10</Mezszám>
        <Szülidő>2003.06.29</Szülidő>

```

```

</Labdarúgó>
<!--ARGENTÍN Labdarúgók-->
<Labdarúgó Lkód="LM10" CSkód="5">
  <Poszt>Középcsatár</Poszt>
  <Poszt>Jobb szélső csatár</Poszt>
  <Név>
    <Keresztnév>Lionel</Keresztnév>
    <Vezetéknév>Messi</Vezetéknév>
  </Név>
  <Gól>9</Gól>
  <Mezszám>10</Mezszám>
  <Szüldő>1987.07.24</Szüldő>
</Labdarúgó>
<Labdarúgó Lkód="EM23" CSkód="5">
  <Poszt>Kapus</Poszt>
  <Név>
    <Keresztnév>Emiliano</Keresztnév>
    <Vezetéknév>Martinez</Vezetéknév>
  </Név>
  <Gól>0</Gól>
  <Mezszám>23</Mezszám>
  <Szüldő>1992.09.02</Szüldő>
</Labdarúgó>
<!--Mérkőzések-->
<Mérkőzés Mkód="1" Tkód="101" Hkód="201">
  <Csapat1Gólok>5</Csapat1Gólok>
  <Csapat2Gólok>2</Csapat2Gólok>
</Mérkőzés>
<Mérkőzés Mkód="2" Tkód="102" Hkód="202">
  <Csapat1Gólok>0</Csapat1Gólok>
  <Csapat2Gólok>1</Csapat2Gólok>
</Mérkőzés>
<Mérkőzés Mkód="3" Tkód="103" Hkód="203">
  <Csapat1Gólok>3</Csapat1Gólok>
  <Csapat2Gólok>7</Csapat2Gólok>
</Mérkőzés>
<Mérkőzés Mkód="4" Tkód="104" Hkód="204">
  <Csapat1Gólok>2</Csapat1Gólok>
  <Csapat2Gólok>1</Csapat2Gólok>
</Mérkőzés>
<Mérkőzés Mkód="5" Tkód="105" Hkód="205">
  <Csapat1Gólok>2</Csapat1Gólok>
  <Csapat2Gólok>3</Csapat2Gólok>
</Mérkőzés>
<!--Helyszínek-->
<Helyszín Hkód="201">
  <Stadion>
    <Név>Education City Stadion</Név>

```

```
        <Férőhely>45000</Férőhely>
        <Tulajdonos>katari kormány</Tulajdonos>
    </Stadion>
    <Város>al-Rajján</Város>
</Helyszín>
<Helyszín Hkód="202">
    <Stadion>
        <Név>Al Bayt Stadion</Név>
        <Férőhely>60000</Férőhely>
        <Tulajdonos>katari kormány</Tulajdonos>
    </Stadion>
    <Város>al-Hor</Város>
</Helyszín>
<Helyszín Hkód="203">
    <Stadion>
        <Név>Loszaíli Nemzeti Stadion</Név>
        <Férőhely>80000</Férőhely>
        <Tulajdonos>katari kormány</Tulajdonos>
    </Stadion>
    <Város>Loszaíl</Város>
</Helyszín>
<Helyszín Hkód="204">
    <Stadion>
        <Név>al-Thumama Stadion</Név>
        <Férőhely>40000</Férőhely>
        <Tulajdonos>katari kormány</Tulajdonos>
    </Stadion>
    <Város>al-Thumama</Város>
</Helyszín>
<Helyszín Hkód="205">
    <Stadion>
        <Név>El-Dzsanúb Stadion</Név>
        <Férőhely>40000</Férőhely>
        <Tulajdonos>katari kormány</Tulajdonos>
    </Stadion>
    <Város>Al-Vakra</Város>
</Helyszín>
<!--Típusok-->
<Típus Tkód="101">
    <Név>Csoportmérkőzés</Név>
</Típus>
<Típus Tkód="102">
    <Név>Nyolcaddöntő</Név>
</Típus>
<Típus Tkód="103">
    <Név>Negyeddöntő</Név>
</Típus>
<Típus Tkód="104">
```

```

        <Név>Elődöntő</Név>
    </Típus>
    <Típus Tkód="105">
        <Név>Döntő</Név>
    </Típus>
    <!--Mérkőzik kapcsolat-->
    <Mérkőzik Mkód="1" Csapat1="1" Csapat2="2" Győztes="1">
        <Dátum>2022.11.22</Dátum>
        <Játékvezető>Frank de Bleeckere</Játékvezető>
    </Mérkőzik>
    <Mérkőzik Mkód="2" Csapat1="3" Csapat2="5" Győztes="5">
        <Dátum>2022.11.26</Dátum>
        <Játékvezető>Pedro Proenca</Játékvezető>
    </Mérkőzik>
    <Mérkőzik Mkód="3" Csapat1="2" Csapat2="1" Győztes="2">
        <Dátum>2022.11.30</Dátum>
        <Játékvezető>Kim Milton Nielsen</Játékvezető>
    </Mérkőzik>
    <Mérkőzik Mkód="4" Csapat1="4" Csapat2="5" Győztes="4">
        <Dátum>2022.12.01</Dátum>
        <Játékvezető>Michel Vautrot</Játékvezető>
    </Mérkőzik>
    <Mérkőzik Mkód="5" Csapat1="1" Csapat2="5" Győztes="1">
        <Dátum>2022.12.18</Dátum>
        <Játékvezető>Oscar Ruiz</Játékvezető>
    </Mérkőzik>
</Világbajnokság_DLWGQZ>

```

4. XMLSchema készítése XML dokumentum alapján

Az XML dokumentumhoz készíteni kellett egy validációt elősegítő sémát. Először kigyűjtöttem az egyszerű típusokat, majd meghatároztam a saját típusokat. Összesen 4 darabot hoztam létre. Például a dátumhoz készítettem egy olyan saját típust, amely reguláris kifejezést (röviden: regex) tartalmaz, illetve enumerációt a jelöléshez. Ezek után elkészítettem a komplex típusokat minden elemre, majd elsődleges- és idegenkulcsokat hoztam létre. A legvégén pedig megvalósítottam az 1:1 kapcsolatot a Csapat és a Logó egyedek között.

XMLSchema forráskódja:

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <!--Egyszerű típusok kigyűjtése-->
    <!--Csapat-->
    <xs:element name="Vezetőedző" type="xs:string" />
    <xs:element name="Név" type="xs:string" />
    <!--Logó-->
    <xs:element name="Színek" type="xs:string" />
    <xs:element name="Tervező" type="xs:string" />

```

```

<!--Csoport-->
<xs:element name="Jelölés" type="jelolesTipus" />
<!--Labdarúgó-->
<xs:element name="Poszt" type="xs:string" />
<xs:element name="Keresztnév" type="xs:string" />
<xs:element name="Vezetéknév" type="xs:string" />
<xs:element name="Gól" type="golTipus" />
<xs:element name="Mezszám" type="xs:int" />
<xs:element name="Szülidő" type="datumTipus" />
<!--Mérkőzés-->
<xs:element name="Csapat1Gólok" type="golTipus" />
<xs:element name="Csapat2Gólok" type="golTipus" />
<!--Helyszín-->
<xs:element name="Férőhely" type="ferohelyTipus" />
<xs:element name="Tulajdonos" type="xs:string" />
<xs:element name="Város" type="xs:string"/>
<!--Mérkőzők-->
<xs:element name="Dátum" type="datumTipus" />
<xs:element name="Játékvezető" type="xs:string" />
<!--Saját típusok meghatározása-->
<xs:simpleType name="jelolesTipus">
  <xs:restriction base="xs:string">
    <xs:enumeration value="A" />
    <xs:enumeration value="B" />
    <xs:enumeration value="C" />
    <xs:enumeration value="D" />
    <xs:enumeration value="E" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="ferohelyTipus">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="30000" />
    <xs:maxInclusive value="100000" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="golTipus">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0" />
    <xs:maxInclusive value="15" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="datumTipus">
  <xs:restriction base="xs:string">
    <xs:pattern value="([12]\d{3}).(0[1-9]|1[0-2]).(0[1-9]|([12]\d|3[01]))"
  />
  </xs:restriction>
</xs:simpleType>
<!--Komplex típusokhoz saját típus meghatározása-->

```

```

<xs:complexType name="csapatTípus">
  <xs:sequence>
    <xs:element ref="Vezetőedző" />
    <xs:element ref="Név" />
  </xs:sequence>
  <xs:attribute name="CSkód" type="xs:integer" use="required" />
  <xs:attribute name="Csapatkapitány" type="xs:string" />
  <xs:attribute name="CSOPkód" type="xs:integer" />
  <xs:attribute name="LOGÓkód" type="xs:integer" />
</xs:complexType>
<xs:complexType name="logóTípus">
  <xs:sequence>
    <xs:element ref="Színek" minOccurs="1" maxOccurs="10"/>
    <xs:element ref="Tervező" />
  </xs:sequence>
  <xs:attribute name="LOGÓkód" type="xs:integer" use="required" />
</xs:complexType>
<xs:complexType name="csoportTípus">
  <xs:sequence>
    <xs:element ref="Jelölés" />
  </xs:sequence>
  <xs:attribute name="CSOPkód" type="xs:integer" use="required" />
</xs:complexType>
<xs:complexType name="labdarúgóTípus">
  <xs:sequence>
    <xs:element ref="Poszt" minOccurs="1" maxOccurs="20"/>
    <xs:element name="Név">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="Keresztnév" />
          <xs:element ref="Vezetéknév" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element ref="Gól" />
    <xs:element ref="Mezszám" />
    <xs:element ref="Szülidő" />
  </xs:sequence>
  <xs:attribute name="Lkód" type="xs:string" />
  <xs:attribute name="CSkód" type="xs:integer" />
</xs:complexType>
<xs:complexType name="mérkőzésTípus">
  <xs:sequence>
    <xs:element ref="Csapat1Gólok" />
    <xs:element ref="Csapat2Gólok" />
  </xs:sequence>
  <xs:attribute name="Mkód" type="xs:integer" />
  <xs:attribute name="Tkód" type="xs:integer" />

```

```

        <xs:attribute name="Hkód" type="xs:integer" />
    </xs:complexType>
    <xs:complexType name="helyszínTípus">
        <xs:sequence>
            <xs:element name="Stadion">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element ref="Név" />
                        <xs:element ref="Férőhely" />
                        <xs:element ref="Tulajdonos" />
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="Város" />
        </xs:sequence>
        <xs:attribute name="Hkód" type="xs:integer" use="required" />
    </xs:complexType>
    <xs:complexType name="típus">
        <xs:sequence>
            <xs:element ref="Név" />
        </xs:sequence>
        <xs:attribute name="Tkód" type="xs:integer" use="required" />
    </xs:complexType>
    <xs:complexType name="mérkőzőikTípus">
        <xs:sequence>
            <xs:element ref="Dátum" />
            <xs:element ref="Játékvezető" />
        </xs:sequence>
        <xs:attribute name="Mkód" type="xs:integer" use="required" />
        <xs:attribute name="Csapat1" type="xs:integer" />
        <xs:attribute name="Csapat2" type="xs:integer" />
        <xs:attribute name="Győztes" type="xs:integer" />
    </xs:complexType>
    <!--Gyökérelemtől az elemek felhasználása-->
    <xs:element name="Világbajnokság_DLWGQZ">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Csapat" type="csapatTípus" minOccurs="1"
maxOccurs="100"/>
                <xs:element name="Logó" type="logóTípus" minOccurs="1"
maxOccurs="100"/>
                <xs:element name="Csoport" type="csoportTípus" minOccurs="1"
maxOccurs="10"/>
                <xs:element name="Labdarúgó" type="labdarúgóTípus" minOccurs="1"
maxOccurs="100"/>
                <xs:element name="Mérkőzés" type="mérkőzésTípus" minOccurs="1"
maxOccurs="100"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

```

```

        <xs:element name="Helyszín" type="helyszínTípus" minOccurs="1"
maxOccurs="100"/>
        <xs:element name="Típus" type="típus" minOccurs="1"
maxOccurs="100"/>
        <xs:element name="Mérkőzik" type="mérkőzikTípus" minOccurs="1"
maxOccurs="100"/>
    </xs:sequence>
</xs:complexType>
<!--Elsődleges kulcsok-->
<xs:key name="csapat_kulcs">
    <xs:selector xpath="Csapat"/>
    <xs:field xpath="@CSkód"/>
</xs:key>
<xs:key name="logó_kulcs">
    <xs:selector xpath="Logó"/>
    <xs:field xpath="@LOGÓkód"/>
</xs:key>
<xs:key name="csoport_kulcs">
    <xs:selector xpath="Csoport"/>
    <xs:field xpath="@CSOPkód"/>
</xs:key>
<xs:key name="labdarúgó_kulcs">
    <xs:selector xpath="Labdarúgó"/>
    <xs:field xpath="@Lkód"/>
</xs:key>
<xs:key name="mérkőzés_kulcs">
    <xs:selector xpath="Mérkőzés"/>
    <xs:field xpath="@Mkód"/>
</xs:key>
<xs:key name="helyszín_kulcs">
    <xs:selector xpath="Helyszín"/>
    <xs:field xpath="@Hkód"/>
</xs:key>
<xs:key name="típus_kulcs">
    <xs:selector xpath="Típus"/>
    <xs:field xpath="@Tkód"/>
</xs:key>
<!--Idegen kulcsok-->
<xs:keyref name="csapat_labdarúgó_kulcs" refer="csapat_kulcs">
    <xs:selector xpath="Labdarúgó"/>
    <xs:field xpath="@CSkód"/>
</xs:keyref>
<xs:keyref name="csapat_mérkőzik_kulcs_csapat1" refer="csapat_kulcs">
    <xs:selector xpath="Mérkőzik"/>
    <xs:field xpath="@Csapat1"/>
</xs:keyref>
<xs:keyref name="csapat_mérkőzik_kulcs_csapat2" refer="csapat_kulcs">
    <xs:selector xpath="Mérkőzik"/>

```



```

        <xs:field xpath="@Csapat2"/>
    </xs:keyref>
    <xs:keyref name="csapat_mérkőzik_kulcs_győztes" refer="csapat_kulcs">
        <xs:selector xpath="Mérkőzik"/>
        <xs:field xpath="@Győztes"/>
    </xs:keyref>
    <xs:keyref name="logó_csapat_kulcs" refer="logó_kulcs">
        <xs:selector xpath="Csapat"/>
        <xs:field xpath="@LOGÓkód"/>
    </xs:keyref>
    <xs:keyref name="csoport_csapat_kulcs" refer="csoport_kulcs">
        <xs:selector xpath="Csapat"/>
        <xs:field xpath="@CSOPkód"/>
    </xs:keyref>
    <xs:keyref name="labdarúgó_csapat_kulcs" refer="labdarúgó_kulcs">
        <xs:selector xpath="Csapat"/>
        <xs:field xpath="@Csapatkapitány"/>
    </xs:keyref>
    <xs:keyref name="mérkőzés_mérkőzik_kulcs" refer="mérkőzés_kulcs">
        <xs:selector xpath="Mérkőzik"/>
        <xs:field xpath="@Mkód"/>
    </xs:keyref>
    <xs:keyref name="helyszín_mérkőzés_kulcs" refer="helyszín_kulcs">
        <xs:selector xpath="Mérkőzés"/>
        <xs:field xpath="@Hkód"/>
    </xs:keyref>
    <xs:keyref name="típus_mérkőzés_kulcs" refer="típus_kulcs">
        <xs:selector xpath="Mérkőzés"/>
        <xs:field xpath="@Tkód"/>
    </xs:keyref>
    <!--Az 1:1 kapcsolat megvalósítás-->
    <xs:unique name="logó_csapat_egyegy">
        <xs:selector xpath="Csapat"/>
        <xs:field xpath="@LOGÓkód"/>
    </xs:unique>
</xs:element>
</xs:schema>

```

Validáció végeredménye: (link: <https://www.freeformatter.com/xml-validator-xsd.html>)

XML Validator – XSD (XML Schema)

Validators / XML Validator – XSD (XML Schema)

Validates the XML string/file against the specified XSD string/file. XSD files are "XML Schemas" that describe the structure of a XML document. The validator checks for well formedness first, meaning that your XML file must be parsable using a DOM/SAX parser, and only then does it validate your XML against the XML Schema. The validator will report fatal errors, non-fatal errors and warnings.

The XML document is valid.



5. DOM program készítése JAVA környezetben

A DOM programokat JAVA-ban készítettem el, ahogy a feladatkiírásban is szerepelt. Az alábbi programokat a következő alfejezetekben fogom részletesebben taglalni.

5a.) DOM adatolvasás

'DOMReadDLWGQZ' osztály: Ez az osztály egy XML fájl elemzésére szolgál a DOM (Document Object Model) technológiát használva. Az osztály több metódust tartalmaz az XML adatok strukturált feldolgozásához.

'main' metódus: Ez a fő belépési pont a program számára, amely inicializálja és elindítja az XML feldolgozást a 'readXmlDocument' metódus segítségével.

'readXMLDocument' metódus: Ez a metódus felelős az XML fájl beolvasásáért, a DOM parser inicializálásáért és a dokumentum struktúrájának normalizálásáért. A különböző XML elemeket külön metódusokkal dolgozza fel, mint például 'readXMLCsapatElement', 'readXMLLogoElement', stb.

'readXmlCsapatElement' metódus: Ez a metódus a „Csapat” elem feldolgozását végzi. Be tudjuk olvasni az elemhez tartozó attribútumokat és gyerekelemeket. Például az attribútumhoz tartozik a csapat kódja, csapatkapitány kódja, csoport kódja és a logó kódja, illetve a gyerekelemekhez pedig a vezetőedző és a csapat neve.

További Almetódusok:

- 'readXMLCsoportElement'
- 'readXMLLabdarugoElement'
- 'readXMLMerkozesElement'
- 'readXMLHelyszinElement'
- 'readXMLTipusElement'
- 'readXMLMerkozikElement'

'DocumentBuilderFactory' és 'DocumentBuilder' osztályok: Ezek az osztályok a DOM (Document Object Model) parser alapvető elemei, amelyeket a dokumentumok létrehozására és olvasására szolgálnak XML formátumban.

'DocumentBuiderFactory' osztály: Segítségével létrehozza a 'DocumentBuilder' objektumokat, amelyek segítségével XML dokumentumokat lehet vele olvasni és létrehozni.

'DocumentBuider' osztály: A 'DocumentBuilder' osztályt a 'DocumentBuilderFactory' objektum segítségével hozzuk létre, illetve a tényleges XML dokumentumok létrehozásáért vagy olvasásáért felelős.

'Node' és 'NodeList' objektumok: A különböző XML elemek (például Csapat, Logó, Csoport, stb.) 'NodeList' objektumokban vannak eltárolva. A 'Node' objektum reprezentálja egy XML dokumentum egy elemét vagy csomópontját a DOM modellben. Az XML fájlban lévő minden entitás (például elem, attribútum, szöveg, stb.) egy 'Node' objektumként jelenik meg.

DOMReadDLWGQZ.java forráskódja:

```
package hu.domparse.dlwqz;

import java.io.File;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import org.w3c.dom.*;

public class DOMReadDLWGQZ {
    // A fő metódus, ami meghívja a readXmlDocument metódust a megadott XML
    // fájlal
    public static void main(String[] args) {
        readXMLDocument("./XMLDLWGQZ.xml");
    }

    // Metódus, amely az XML fájl beolvasására és feldolgozására szolgál
    private static void readXMLDocument(String filePath) {
        try {
            File xmlFile = new File("XMLDLWGQZ.xml");
            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
            DocumentBuilder dBuilder = factory.newDocumentBuilder();
            Document document = dBuilder.parse(xmlFile);
            document.getDocumentElement().normalize();

            // Kiírjuk a gyökérelem nevét a konzolra
            System.out.println("Gyökérelem: " +
                document.getDocumentElement().getNodeName());

            // XML elemek feldolgozása külön metódusokkal
            readXMLCsapatElement(document, "Csapat");
            readXMLLogoElement(document, "Logó");
            readXMLCsoportElement(document, "Csoport");
            readXMLLabdarugoElement(document, "Labdarúgó");
            readXMLMerkozesElement(document, "Mérkőzés");
            readXMLHelyszinElement(document, "Helyszín");
            readXMLTipusElement(document, "Típus");
            readXMLMerkozikElement(document,
                "Mérkőzik");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    // Csapat elemek feldolgozása
    private static void readXMLCsapatElement(Document document, String
        elementName) {
        System.out.println("\n-----CSAPAT-ELEM-----");
    }
}
```

```

        NodeList csapatNodeList = document.getElementsByTagName(elementName);

        for(int i = 0; i < csapatNodeList.getLength(); i++) {
            Node csapatNode = csapatNodeList.item(i);
            System.out.println("\nJelenlegi elem: " + csapatNode.getNodeName());

            // Ellenőrzi, hogy a Node egy ELEMENT_NODE típusú-e
            if(csapatNode.getNodeType() == Node.ELEMENT_NODE) {
                Element csapatElem = (Element) csapatNode;

                // attribútumok létrehozása és megadása
                String cskod = csapatElem.getAttribute("CSkód");
                String csapatkapitany = csapatElem.getAttribute("Csapatkapitány");
                String csopkod = csapatElem.getAttribute("CSOPkód");
                String logokod = csapatElem.getAttribute("LOGÓkód");

                Node vezetoedzoNode =
csapatElem.getElementsByTagName("Vezetőedző").item(0);
                String vezetoedzo = vezetoedzoNode.getTextContent();
                Node csapatnevNode =
csapatElem.getElementsByTagName("Név").item(0);
                String csapatnev = csapatnevNode.getTextContent();

                // Kiíjuk a konzolra az attribútumokat és az elemeket
                System.out.println("Cskód: " + cskod);
                System.out.println("Csapatkapitány: " + csapatkapitany);
                System.out.println("CSOPkód: " + csopkod);
                System.out.println("LOGÓkód: " + logokod);
                System.out.println("Vezetőedző: " + vezetoedzo);
                System.out.println("Csapatnév: " + csapatnev);
                System.out.println("-----");
            }
        }
    }

    // Logó elemek feldolgozása
    private static void readXMLLogoElement(Document document, String elementName)
    {
        System.out.println("\n-----LOGÓ-ELEM-----");
        NodeList logoNodeList = document.getElementsByTagName(elementName);

        for(int i = 0; i < logoNodeList.getLength(); i++) {
            Node logoNode = logoNodeList.item(i);
            System.out.println("\nJelenlegi elem: " + logoNode.getNodeName());

            // Ellenőrzi, hogy a Node egy ELEMENT_NODE típusú-e
            if(logoNode.getNodeType() == Node.ELEMENT_NODE) {
                Element logoElem = (Element) logoNode;

```

```

        // attribútum létrehozása és megadása
        String logokod = logoElem.getAttribute("LOGÓkód");
        System.out.println("LOGÓkód: " + logokod);

        //Ellenőrzi, hogy a gyerekelemek hossza nagyobb-e, mint 3
        if(logoNodeList.item(i).getChildNodes().getLength() > 3) {
            int darab = 0;
            Node szinekNode =
logoElem.getElementsByTagName("Színek").item(0);
            while(szinekNode != null) {
                szinekNode =
logoElem.getElementsByTagName("Színek").item(darab);
                if(szinekNode != null) {
                    String szinek = szinekNode.getTextContent();
                    System.out.println("A logó színe: " + szinek);
                }
                darab++;
            }
        }

        Node tervezoNode =
logoElem.getElementsByTagName("Tervező").item(0);
        String tervezo = tervezoNode.getTextContent();

        System.out.println("Tervező: " + tervezo);
        System.out.println("-----");
    }
}

// Csoport elemek feldolgozása
private static void readXMLCsoportElement(Document document, String
elementName) {
    System.out.println("\n-----CSOPORT-ELEM-----");
    NodeList csoportNodeList = document.getElementsByTagName(elementName);

    for(int i = 0; i < csoportNodeList.getLength(); i++) {
        Node csoportNode = csoportNodeList.item(i);
        System.out.println("\nJelenlegi elem: " + csoportNode.getNodeName());

        // Ellenőrzi, hogy a Node egy ELEMENT_NODE típusú-e
        if(csoportNode.getNodeType() == Node.ELEMENT_NODE) {
            // attribútum létrehozása és megadása
            Element csoportElem = (Element) csoportNode;
            String csopkod = csoportElem.getAttribute("CSOPkód");
            System.out.println("CSOPkód: " + csopkod);
        }
    }
}

```

```

        Node jelolesNode =
csoportElem.getElementsByTagName("Jelölés").item(0);
        String jeloles = jelolesNode.getTextContent();

        System.out.println("A csoport jelölése: " + jeloles);
        System.out.println("-----");
    }
}

// Labdarúgó elemek feldolgozása
private static void readXMLLabdarugoElement(Document document, String
elementName) {
    System.out.println("\n-----LABDARÚGÓ-ELEM-----");
    NodeList labdarugoNodeList = document.getElementsByTagName(elementName);

    for(int i = 0; i < labdarugoNodeList.getLength(); i++) {
        Node labdarugoNode = labdarugoNodeList.item(i);
        System.out.println("\nJelenlegi elem: " +
labdarugoNode.getNodeName());

        // Ellenőrzi, hogy a Node egy ELEMENT_NODE típusú-e
        if(labdarugoNode.getNodeType() == Node.ELEMENT_NODE) {
            // attribútum létrehozása és megadása
            Element labdarugoElem = (Element) labdarugoNode;
            String lkod = labdarugoElem.getAttribute("Lkód");
            System.out.println("Lkód: " + lkod);
            String cskod = labdarugoElem.getAttribute("CSkód");
            System.out.println("CSkód: " + cskod);

            //Ellenőrzi, hogy a gyerekelemek hossza nagyobb-e, mint 3
            if(labdarugoNodeList.item(i).getChildNodes().getLength() > 3) {
                int darab = 0;
                Node posztNode =
labdarugoElem.getElementsByTagName("Poszt").item(0);
                while(posztNode != null) {
                    posztNode =
labdarugoElem.getElementsByTagName("Poszt").item(darab);
                    if(posztNode != null) {
                        String poszt = posztNode.getTextContent();
                        System.out.println("A labdarúgó posztja: " + poszt);
                    }
                    darab++;
                }
            }

            //Ellenőrzi, hogy a gyerekelemek hossza nagyobb-e, mint 3
            if(labdarugoNodeList.item(i).getChildNodes().getLength() > 3) {

```

```

        int darab = 0;
        Node nevNode =
labdarugoElem.getElementsByTagName("Név").item(0);
        while(nevNode != null) {
            nevNode =
labdarugoElem.getElementsByTagName("Név").item(darab);
            if(nevNode != null) {
                Node keresztnevNode =
labdarugoElem.getElementsByTagName("Keresztnév").item(darab);
                String keresztnev = keresztnevNode.getTextContent();
                System.out.println("A labdarúgó keresztneve: " +
keresztnev);

                Node vezeteknevNode =
labdarugoElem.getElementsByTagName("Vezetéknév").item(darab);
                String vezeteknev = vezeteknevNode.getTextContent();
                System.out.println("A labdarúgó vezetékeve: " +
vezeteknev);

            }
            darab++;
        }
    }

    Node golNode = labdarugoElem.getElementsByTagName("Gól").item(0);
    String gol = golNode.getTextContent();
    System.out.println("Gólok száma: " + gol);
    Node mezzsamNode =
labdarugoElem.getElementsByTagName("Mezszám").item(0);
    String mezzsam = mezzsamNode.getTextContent();
    System.out.println("Mezszám: " + mezzsam);
    Node szulidoNode =
labdarugoElem.getElementsByTagName("Szülidő").item(0);
    String szulido = szulidoNode.getTextContent();
    System.out.println("Születési dátum: " + szulido);
    System.out.println("-----");
}
}

// Mérkőzés elemek feldolgozása
private static void readXMLMerkozesElement(Document document, String
elementName) {
    System.out.println("\n-----MÉRKŐZÉS-ELEM-----");
    NodeList merkozesNodeList = document.getElementsByTagName("Mérkőzés");
    for(int i = 0; i < merkozesNodeList.getLength(); i++) {
        Node merkozesNode = merkozesNodeList.item(i);
        System.out.println("\nJelenlegi elem: " + merkozesNode.getNodeName());

        // Ellenőrzi, hogy a Node egy ELEMENT_NODE típusú-e

```

```

        if(merkozesNode.getNodeType() == Node.ELEMENT_NODE) {
            // attribútum létrehozása és megadása
            Element merkozesElem = (Element) merkozesNode;
            String mkod = merkozesElem.getAttribute("Mkód");
            System.out.println("Mkód: " + mkod);
            String tkod = merkozesElem.getAttribute("Tkód");
            System.out.println("Tkód: " + tkod);
            String hkod = merkozesElem.getAttribute("Hkód");
            System.out.println("Hkód: " + hkod);

            Node csapat1Node =
merkozesElem.getElementsByTagName("Csapat1Gólok").item(0);
            String csapat1Gol = csapat1Node.getTextContent();
            System.out.println("Csapat1 góljainak száma: " + csapat1Gol);
            Node csapat2Node =
merkozesElem.getElementsByTagName("Csapat2Gólok").item(0);
            String csapat2Gol = csapat2Node.getTextContent();
            System.out.println("Csapat2 góljainak száma: " + csapat2Gol);
            System.out.println("-----");
        }
    }

    // Helyszín elemek feldolgozása
    private static void readXMLHelyszinElement(Document document, String
elementName) {
        System.out.println("\n-----HELYSZÍN-ELEM-----");
        NodeList helyszinNodeList = document.getElementsByTagName("Helyszín");

        for(int i = 0; i < helyszinNodeList.getLength(); i++) {
            Node helyszinNode = helyszinNodeList.item(i);
            System.out.println("\nJelenlegi elem: " + helyszinNode.getNodeName());

            // Ellenőrzi, hogy a Node egy ELEMENT_NODE típusú-e
            if(helyszinNode.getNodeType() == Node.ELEMENT_NODE) {
                // attribútum létrehozása és megadása
                Element helyszinElem = (Element) helyszinNode;
                String hkod = helyszinElem.getAttribute("Hkód");
                System.out.println("Hkód: " + hkod);

                //Ellenőrzi, hogy a gyerekelemek hossza nagyobb-e, mint 3
                if(helyszinNodeList.item(i).getChildNodes().getLength() > 3) {
                    int darab = 0;
                    Node stadionNode =
helyszinElem.getElementsByTagName("Stadion").item(0);
                    while(stadionNode != null) {
                        stadionNode =
helyszinElem.getElementsByTagName("Stadion").item(darab);

```



```

        if(stadionNode != null) {
            Node nevNode =
helyszinElem.getElementsByTagName("Név").item(darab);
            String nev = nevNode.getTextContent();
            System.out.println("A stadion neve: " + nev);
            Node ferohelyNode =
helyszinElem.getElementsByTagName("Férőhely").item(darab);
            String ferohely = ferohelyNode.getTextContent();
            System.out.println("Férőhely: " + ferohely + " fő");
            Node tulajdonosNode =
helyszinElem.getElementsByTagName("Tulajdonos").item(darab);
            String tulajdonos = tulajdonosNode.getTextContent();
            System.out.println("Tulajdonos: " + tulajdonos);
        }
        darab++;
    }
}

Node varosNode =
helyszinElem.getElementsByTagName("Város").item(0);
String varos = varosNode.getTextContent();
System.out.println("Város: " + varos);
System.out.println("-----");
}
}

// Típus elemek feldolgozása
private static void readXMLTipusElement(Document document, String elementName)
{
    System.out.println("\n-----TÍPUS-ELEM-----");
    NodeList tipusNodeList = document.getElementsByTagName("Típus");

    for(int i = 0; i < tipusNodeList.getLength(); i++) {
        Node tipusNode = tipusNodeList.item(i);
        System.out.println("\nJelenlegi elem: " + tipusNode.getNodeName());

        // Ellenőrzi, hogy a Node egy ELEMENT_NODE típusú-e
        if(tipusNode.getNodeType() == Node.ELEMENT_NODE) {
            // attribútum létrehozása és megadása
            Element tipusElem = (Element) tipusNode;
            String tkod = tipusElem.getAttribute("Tkód");
            System.out.println("Tkód: " + tkod);

            Node nevNode = tipusElem.getElementsByTagName("Név").item(0);
            String nev = nevNode.getTextContent();
            System.out.println("Mérkőzés típusa: " + nev);
            System.out.println("-----");
        }
    }
}

```

```

    }
}

// Mérkőzik elemek feldolgozása
private static void readXMLMerkozikElement(Document document, String
elementName) {
    System.out.println("\n-----MÉRKŐZIK-ELEM-----");
    NodeList merkozikNodeList = document.getElementsByTagName("Mérkőzik");

    for(int i = 0; i < merkozikNodeList.getLength(); i++) {
        Node merkozikNode = merkozikNodeList.item(i);
        System.out.println("\nJelenlegi elem: " + merkozikNode.getNodeName());

        // Ellenőrzi, hogy a Node egy ELEMENT_NODE típusú-e
        if(merkozikNode.getNodeType() == Node.ELEMENT_NODE) {
            // attribútum létrehozása és megadása
            Element merkozikElem = (Element) merkozikNode;
            String mkod = merkozikElem.getAttribute("Mkód");
            System.out.println("Mkód: " + mkod);
            String csapat1 = merkozikElem.getAttribute("Csapat1");
            System.out.println("Csapat1 kódja: " + csapat1);
            String csapat2 = merkozikElem.getAttribute("Csapat2");
            System.out.println("Csapat2 kódja: " + csapat2);
            String gyoztes = merkozikElem.getAttribute("Győztes");
            System.out.println("Győztes csapat kódja: " + gyoztes);

            Node datumNode =
merkozikElem.getElementsByTagName("Dátum").item(0);
            String datum = datumNode.getTextContent();
            System.out.println("Dátum: " + datum);
            Node jatekvezetoNode =
merkozikElem.getElementsByTagName("Játékvezető").item(0);
            String jatekvezeto = jatekvezetoNode.getTextContent();
            System.out.println("Játékvezető: " + jatekvezeto);
            System.out.println("-----");
        }
    }
}
}

```

Output:

Gyökérelem: Világbajnokság_DLWGQZ

-----CSAPAT-ELEM-----

Jelenlegi elem: Csapat

Cskód: 1

Csapatkapitány: CR7
CSOPkód: 11
LOGÓkód: 21
Vezetőedző: Roberto Martínez
Csapatnév: Portugália

Jelenlegi elem: Csapat
Cskód: 2
Csapatkapitány: TS1
CSOPkód: 12
LOGÓkód: 22
Vezetőedző: Hansi Flick
Csapatnév: Németország

Jelenlegi elem: Csapat
Cskód: 3
Csapatkapitány: KM10
CSOPkód: 13
LOGÓkód: 23
Vezetőedző: Didier Deschamp
Csapatnév: Franciaország

Jelenlegi elem: Csapat
Cskód: 4
Csapatkapitány: HK9
CSOPkód: 14
LOGÓkód: 24
Vezetőedző: Gareth Southgate
Csapatnév: Anglia

Jelenlegi elem: Csapat
Cskód: 5
Csapatkapitány: LM10
CSOPkód: 15
LOGÓkód: 25
Vezetőedző: Lionel Scaloni
Csapatnév: Argentína

-----LOGÓ-ELEM-----

Jelenlegi elem: Logó
LOGÓkód: 21
A logó színe: Piros
A logó színe: Kék
A logó színe: Zöld
Tervező: António Modesto

Jelenlegi elem: Logó

LOGÓkód: 22
A logó színe: Fekete
A logó színe: Piros
A logó színe: Arany
Tervező: Anton Stankowski

Jelenlegi elem: Logó
LOGÓkód: 23
A logó színe: Kék
A logó színe: Fehér
A logó színe: Piros
Tervező: Raymond Savignac

Jelenlegi elem: Logó
LOGÓkód: 24
A logó színe: Fehér
A logó színe: Kék
A logó színe: Piros
Tervező: William C. Gibbons

Jelenlegi elem: Logó
LOGÓkód: 25
A logó színe: Égkék
A logó színe: Fehér
A logó színe: Sárga
Tervező: Salvador Dellutri

-----CSOPORT-ELEM-----

Jelenlegi elem: Csoport
CSOPkód: 11
A csoport jelölése: A

Jelenlegi elem: Csoport
CSOPkód: 12
A csoport jelölése: B

Jelenlegi elem: Csoport
CSOPkód: 13
A csoport jelölése: C

Jelenlegi elem: Csoport
CSOPkód: 14
A csoport jelölése: D

Jelenlegi elem: Csoport

CSOPkód: 15

A csoport jelölése: E

-----LABDARÚGÓ-ELEM-----

Jelenlegi elem: Labdarúgó

Lkód: CR7

CSkód: 1

A labdarúgó posztja: Középcsatár

A labdarúgó keresztnéve: Cristiano

A labdarúgó vezetéknéve: Ronaldo

Gólok száma: 10

Mezszám: 7

Születési dátum: 1985.02.05

Jelenlegi elem: Labdarúgó

Lkód: RN18

CSkód: 1

A labdarúgó posztja: Középpályás

A labdarúgó posztja: Védekező középpályás

A labdarúgó keresztnéve: Ruben

A labdarúgó vezetéknéve: Neves

Gólok száma: 2

Mezszám: 18

Születési dátum: 1997.03.13

Jelenlegi elem: Labdarúgó

Lkód: TS1

CSkód: 2

A labdarúgó posztja: Kapus

A labdarúgó keresztnéve: Marc-André

A labdarúgó vezetéknéve: ter Stegen

Gólok száma: 0

Mezszám: 22

Születési dátum: 1992.04.05

Jelenlegi elem: Labdarúgó

Lkód: JK6

CSkód: 2

A labdarúgó posztja: Középpályás

A labdarúgó posztja: Jobb szélső védő

A labdarúgó keresztnéve: Joshua

A labdarúgó vezetéknéve: Kimmich

Gólok száma: 3

Mezszám: 6

Születési dátum: 1995.02.08

Jelenlegi elem: Labdarúgó

Lkód: KM10

CSkód: 3
A labdarúgó posztja: Középcsatár
A labdarúgó keresztnéve: Kylian
A labdarúgó vezetéknéve: Mbappé
Gólok száma: 9
Mezszám: 10
Születési dátum: 1998.12.20

Jelenlegi elem: Labdarúgó
Lkód: EC25
CSkód: 3
A labdarúgó posztja: Középpályás
A labdarúgó posztja: Védekező középpályás
A labdarúgó posztja: Bal szélső védő
A labdarúgó keresztnéve: Eduardo
A labdarúgó vezetéknéve: Camavinga
Gólok száma: 3
Mezszám: 25
Születési dátum: 2002.11.10

Jelenlegi elem: Labdarúgó
Lkód: HK9
CSkód: 4
A labdarúgó posztja: Középcsatár
A labdarúgó keresztnéve: Harry
A labdarúgó vezetéknéve: Kane
Gólok száma: 7
Mezszám: 9
Születési dátum: 1993.07.28

Jelenlegi elem: Labdarúgó
Lkód: JB10
CSkód: 4
A labdarúgó posztja: Középpályás
A labdarúgó keresztnéve: Jude
A labdarúgó vezetéknéve: Bellingham
Gólok száma: 6
Mezszám: 10
Születési dátum: 2003.06.29

Jelenlegi elem: Labdarúgó
Lkód: LM10
CSkód: 5
A labdarúgó posztja: Középcsatár
A labdarúgó posztja: Jobb szélső csatár
A labdarúgó keresztnéve: Lionel
A labdarúgó vezetéknéve: Messi
Gólok száma: 9
Mezszám: 10
Születési dátum: 1987.07.24

Jelenlegi elem: Labdarúgó
Lkód: EM23
CSkód: 5
A labdarúgó posztja: Kapus
A labdarúgó keresztnéve: Emiliano
A labdarúgó vezetéknéve: Martinez
Gólok száma: 0
Mezszám: 23
Születési dátum: 1992.09.02

-----MÉRKŐZÉS-ELEM-----

Jelenlegi elem: Mérkőzés
Mkód: 1
Tkód: 101
Hkód: 201
Csapat1 góljainak száma: 5
Csapat2 góljainak száma: 2

Jelenlegi elem: Mérkőzés
Mkód: 2
Tkód: 102
Hkód: 202
Csapat1 góljainak száma: 0
Csapat2 góljainak száma: 1

Jelenlegi elem: Mérkőzés
Mkód: 3
Tkód: 103
Hkód: 203
Csapat1 góljainak száma: 3
Csapat2 góljainak száma: 7

Jelenlegi elem: Mérkőzés
Mkód: 4
Tkód: 104
Hkód: 204
Csapat1 góljainak száma: 2
Csapat2 góljainak száma: 1

Jelenlegi elem: Mérkőzés
Mkód: 5
Tkód: 105
Hkód: 205
Csapat1 góljainak száma: 2
Csapat2 góljainak száma: 3

-----HELYSZÍN-ELEM-----

Jelenlegi elem: Helyszín
Hkód: 201
A stadion neve: Education City Stadion
Férőhely: 45000 fő
Tulajdonos: katari kormány
Város: al-Rajján

Jelenlegi elem: Helyszín
Hkód: 202
A stadion neve: Al Bayt Stadion
Férőhely: 60000 fő
Tulajdonos: katari kormány
Város: al-Hor

Jelenlegi elem: Helyszín
Hkód: 203
A stadion neve: Loszaíli Nemzeti Stadion
Férőhely: 80000 fő
Tulajdonos: katari kormány
Város: Loszaíl

Jelenlegi elem: Helyszín
Hkód: 204
A stadion neve: al-Thumama Stadion
Férőhely: 40000 fő
Tulajdonos: katari kormány
Város: al-Thumama

Jelenlegi elem: Helyszín
Hkód: 205
A stadion neve: El-Dzsanúb Stadion
Férőhely: 40000 fő
Tulajdonos: katari kormány
Város: Al-Vakra

-----TÍPUS-ELEM-----

Jelenlegi elem: Típus
Tkód: 101
Mérkőzés típusa: Csoportmérkőzés

Jelenlegi elem: Típus
Tkód: 102
Mérkőzés típusa: Nyolcaddöntő

Jelenlegi elem: Típus
Tkód: 103
Mérkőzés típusa: Negyedöntő

Jelenlegi elem: Típus
Tkód: 104
Mérkőzés típusa: Elődöntő

Jelenlegi elem: Típus
Tkód: 105
Mérkőzés típusa: Döntő

-----MÉRKŐZIK-ELEM-----

Jelenlegi elem: Mérkőzik
Mkód: 1
Csapat1 kódja: 1
Csapat2 kódja: 2
Győztes csapat kódja: 1
Dátum: 2022.11.22
Játékvezető: Frank de Bleeckere

Jelenlegi elem: Mérkőzik
Mkód: 2
Csapat1 kódja: 3
Csapat2 kódja: 5
Győztes csapat kódja: 5
Dátum: 2022.11.26
Játékvezető: Pedro Proenca

Jelenlegi elem: Mérkőzik
Mkód: 3
Csapat1 kódja: 2
Csapat2 kódja: 1
Győztes csapat kódja: 2
Dátum: 2022.11.30
Játékvezető: Kim Milton Nielsen

Jelenlegi elem: Mérkőzik
Mkód: 4
Csapat1 kódja: 4
Csapat2 kódja: 5
Győztes csapat kódja: 4
Dátum: 2022.12.01
Játékvezető: Michel Vautrot

Jelenlegi elem: Mérkőzik

Mkód: 5

Csapat1 kódja: 1

Csapat2 kódja: 5

Győztes csapat kódja: 1

Dátum: 2022.12.18

Játékvezető: Oscar Ruiz

5b.) DOM adatlekérdezés

'DOMQueryDLWGQZ' osztály: Ez az osztály olyan metódusokat tartalmaz, amelyek egy XML fájl tartalmát képes beolvasni és feldolgozni DOM segítségével, ráadásul lekérdezéseket is képesek vagyunk végrehajtani.

'queryXMLDocument' metódus: XML fájl beolvasására és feldolgozására szolgál. Paraméterként egy String típust vár, amelyben megadjuk a fájl elérési útját.

Összesen 5 darab lekérdezést készítettem el.

Metódusok és a feladatleírás:

1. **'queryXMLAllFootballersData':** Az összes labdarúgó adatainak kiírása.
2. **'queryXMLAllTeamsName':** Az összes csapat nevének kiírása.
3. **'queryXMLAllMatchesDateAndReferee':** Mérkőzés dátumának és játékvezetőjének kiírása.
4. **'queryXMLDesignerNameWithID21':** '21'-es ID-val rendelkező logó tervezőjének neve.
5. **'queryXMLAllStadiumsNameEquals40000':** Az összes stadion nevének kiírása, amelynek férőhelyeinek száma pontosan 40000 fő.

DOMQueryDLWGQZ.java forráskódja:

```
package hu.domparse.dlwqz;

import java.io.File;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import org.w3c.dom.*;

public class DOMQueryDLWGQZ {
    // A fő metódus, ami meghívja a queryXMLDocument metódust a megadott XML
    // fájlal
    public static void main(String[] args) {
        // Metódus meghívása
        queryXMLDocument("./XMLDLWGQZ.xml");
    }

    // Metódus, amely az XML fájl beolvasására és feldolgozására szolgál
    private static void queryXMLDocument(String filePath) {
        try {
            File xmlFile = new File("XMLDLWGQZ.xml");
```

```

DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
DocumentBuilder dBuilder = factory.newDocumentBuilder();
Document document = dBuilder.parse(xmlFile);
document.getDocumentElement().normalize();

//1. Az összes labdarúgó adatainak kiírása
queryXMLAllFootballersData(document, "Labdarúgó");
//2. Az összes csapat neve
queryXMLAllTeamsName(document, "Csapat");
//3. Mérkőzés dátuma és játékvezető kiírása
queryXMLAllMatchesDateAndReferee(document, "Csapat");
//4. '21'-es ID-val rendelkező logó tervezőjének neve
queryXMLDesignerNameWithID21(document, "Logó");
//5. Az összes stadion nevének kiírása, amelynek a férőhelyeinek száma
    pontosan 40000 fő
        queryXMLAllStadiumsNameEquals40000(document, "Helyszín");

    } catch (Exception e) {
        e.printStackTrace();
    }
}

private static void queryXMLAllStadiumsNameEquals40000(Document document,
String elementName) {
    System.out.println("\n--5.-Az-összes-stadion-nevének-kiírása,-amelynek-a-
férőhelyeinek-pontosan-40000-fő--");
    NodeList helyszinNodeList = document.getElementsByTagName("Helyszín");

    for (int i = 0; i < helyszinNodeList.getLength(); i++) {
        Node helyszinNode = helyszinNodeList.item(i);

        if (helyszinNode.getNodeType() == Node.ELEMENT_NODE) {
            Element helyszinElement = (Element) helyszinNode;
            String stadionNev =
helyszinElement.getElementsByTagName("Név").item(0).getTextContent();
            int ferohely =
Integer.parseInt(helyszinElement.getElementsByTagName("Férőhely").item(0).getTextC
ontent());

            if (ferohely == 40000) {
                System.out.println("Stadion neve: " + stadionNev);
                System.out.println("-----");
            }
        }
    }
}
}

```

```

        private static void queryXMLDesignerNameWithID21(Document document, String
elementName) {
            System.out.println("\n--4.-'21'-es-ID-val-rendelkező-logó-tervezőjének-
neve--");
            String logokod = "21";
            NodeList logoNodeList = document.getElementsByTagName(elementName);
            for(int i = 0; i < logoNodeList.getLength(); i++) {
                Node logoNode = logoNodeList.item(i);

                if(logoNode.getNodeType() == Node.ELEMENT_NODE) {
                    Element logoElem = (Element) logoNode;
                    String logokodAttribute = logoElem.getAttribute("LOGÓkód");
                    if(logokodAttribute.equals(logokod)) {
                        String tervezo =
logoElem.getElementsByTagName("Tervező").item(0).getTextContent();
                        System.out.println("A '" + logokod + "'-es ID-val rendelkező
logó tervezője: " + tervezo);
                        System.out.println("-----
-----");
                    }
                }
            }
        }
    }
}

```

```

        private static void queryXMLAllMatchesDateAndReferee(Document document, String
string) {
            System.out.println("\n--3.-Mérkőzés-dátuma-és-játékvezető-kiírása--");
            NodeList merkozikNodeList = document.getElementsByTagName("Mérkőzik");

            for(int i = 0; i < merkozikNodeList.getLength(); i++) {
                Node merkozikNode = merkozikNodeList.item(i);

                Element merkozikElem = (Element) merkozikNode;

                Node datumNode = merkozikElem.getElementsByTagName("Dátum").item(0);
                String datum = datumNode.getTextContent();
                System.out.println("A mérkőzés dátuma: " + datum);
                Node jatekvezetoNode =
merkozikElem.getElementsByTagName("Játékvezető").item(0);
                String jatekvezeto = jatekvezetoNode.getTextContent();
                System.out.println("A játékvezető neve: " + jatekvezeto);
                System.out.println("-----");
            }
        }
    }
}

```

```

        private static void queryXMLAllTeamsName(Document document, String
elementName) {

```

```

System.out.println("\n--2.-Az-összes-csapat-neve--");
NodeList csapatNodeList = document.getElementsByTagName(elementName);

for(int i = 0; i < csapatNodeList.getLength(); i++) {
    Node csapatNode = csapatNodeList.item(i);

    if(csapatNode.getNodeType() == Node.ELEMENT_NODE &&
csapatNode.getNodeName().equals("Csapat")) {
        Element csapatElem = (Element) csapatNode;

        Node csapatnevNode =
csapatElem.getElementsByTagName("Név").item(0);
        String csapatnev = csapatnevNode.getTextContent();
        System.out.println("Csapat neve: " + csapatnev);
        System.out.println("-----");
    }
}

private static void queryXMLAllFootballersData(Document document, String
elementName) {
    System.out.println("--1.-Az-összes-labdarúgó-adatainak-kiírása--");
    NodeList labdarugoNodeList = document.getElementsByTagName(elementName);

    for(int i = 0; i < labdarugoNodeList.getLength(); i++) {
        Node labdarugoNode = labdarugoNodeList.item(i);

        if(labdarugoNode.getNodeType() == Node.ELEMENT_NODE &&
labdarugoNode.getNodeName().equals("Labdarúgó")) {
            // attribútum létrehozása és megadása
            Element labdarugoElem = (Element) labdarugoNode;
            String lkod = labdarugoElem.getAttribute("Lkód");
            System.out.println("A labdarúgó kódja: " + lkod);
            String cskod = labdarugoElem.getAttribute("CSKód");
            System.out.println("A csapat kódja: " + cskod);

            //Ellenőrizzük, hogy a többszörösen előforduló elemből több van-e,
mint 3 darab
            if(labdarugoNodeList.item(i).getChildNodes().getLength() > 3) {
                int darab = 0;
                Node posztNode =
labdarugoElem.getElementsByTagName("Poszt").item(0);
                while(posztNode != null) {
                    posztNode =
labdarugoElem.getElementsByTagName("Poszt").item(darab);
                    if(posztNode != null) {
                        String poszt = posztNode.getTextContent();
                        System.out.println("A labdarúgó posztja: " + poszt);
                    }
                }
            }
        }
    }
}

```

```

        }
        darab++;
    }
}

//Ellenőrizzük, hogy a többszörösen előforduló elemből több van-e,
mint 3 darab
if(labdarugoNodeList.item(i).getChildNodes().getLength() > 3) {
    int darab = 0;
    Node nevNode =
labdarugoElem.getElementsByTagName("Név").item(0);
    while(nevNode != null) {
        nevNode =
labdarugoElem.getElementsByTagName("Név").item(darab);
        if(nevNode != null) {
            Node keresztnevNode =
labdarugoElem.getElementsByTagName("Keresztnév").item(darab);
            String keresztnev = keresztnevNode.getTextContent();
            System.out.println("A labdarúgó keresztnéve: " +
keresztnev);

            Node vezeteknevNode =
labdarugoElem.getElementsByTagName("Vezetéknév").item(darab);
            String vezeteknev = vezeteknevNode.getTextContent();
            System.out.println("A labdarúgó vezetéknéve: " +
vezeteknev);

        }
        darab++;
    }
}

Node golNode = labdarugoElem.getElementsByTagName("Gól").item(0);
String gol = golNode.getTextContent();
System.out.println("Gólok száma: " + gol);
Node mezzsamNode =
labdarugoElem.getElementsByTagName("Mezszám").item(0);
String mezzsam = mezzsamNode.getTextContent();
System.out.println("Mezszám: " + mezzsam);
Node szulidoNode =
labdarugoElem.getElementsByTagName("Szülidő").item(0);
String szulido = szulidoNode.getTextContent();
System.out.println("Születési dátum: " + szulido);
System.out.println("-----");
}
}
}
}

```

Output:

--1.-Az-összes-labdarúgó-adatainak-kiírása--

A labdarúgó kódja: CR7

A csapat kódja: 1

A labdarúgó posztja: Középcsatár

A labdarúgó keresztnéve: Cristiano

A labdarúgó vezetéknéve: Ronaldo

Gólok száma: 10

Mezszám: 7

Születési dátum: 1985.02.05

A labdarúgó kódja: RN18

A csapat kódja: 1

A labdarúgó posztja: Középpályás

A labdarúgó posztja: Védekező középpályás

A labdarúgó keresztnéve: Ruben

A labdarúgó vezetéknéve: Neves

Gólok száma: 2

Mezszám: 18

Születési dátum: 1997.03.13

A labdarúgó kódja: TS1

A csapat kódja: 2

A labdarúgó posztja: Kapus

A labdarúgó keresztnéve: Marc-André

A labdarúgó vezetéknéve: ter Stegen

Gólok száma: 0

Mezszám: 22

Születési dátum: 1992.04.05

A labdarúgó kódja: JK6

A csapat kódja: 2

A labdarúgó posztja: Középpályás

A labdarúgó posztja: Jobb szélső védő

A labdarúgó keresztnéve: Joshua

A labdarúgó vezetéknéve: Kimmich

Gólok száma: 3

Mezszám: 6

Születési dátum: 1995.02.08

A labdarúgó kódja: KM10

A csapat kódja: 3

A labdarúgó posztja: Középcsatár

A labdarúgó keresztnéve: Kylian

A labdarúgó vezetéknéve: Mbappé

Gólok száma: 9

Mezszám: 10

Születési dátum: 1998.12.20

A labdarúgó kódja: EC25

A csapat kódja: 3

A labdarúgó posztja: Középpályás

A labdarúgó posztja: Védekező középpályás

A labdarúgó posztja: Bal szélső védő

A labdarúgó keresztnéve: Eduardo
A labdarúgó vezetékneve: Camavinga
Gólok száma: 3
Mezszám: 25
Születési dátum: 2002.11.10

A labdarúgó kódja: HK9
A csapat kódja: 4
A labdarúgó posztja: Középcsatár
A labdarúgó keresztnéve: Harry
A labdarúgó vezetékneve: Kane
Gólok száma: 7
Mezszám: 9
Születési dátum: 1993.07.28

A labdarúgó kódja: JB10
A csapat kódja: 4
A labdarúgó posztja: Középpályás
A labdarúgó keresztnéve: Jude
A labdarúgó vezetékneve: Bellingham
Gólok száma: 6
Mezszám: 10
Születési dátum: 2003.06.29

A labdarúgó kódja: LM10
A csapat kódja: 5
A labdarúgó posztja: Középcsatár
A labdarúgó posztja: Jobb szélső csatár
A labdarúgó keresztnéve: Lionel
A labdarúgó vezetékneve: Messi
Gólok száma: 9
Mezszám: 10
Születési dátum: 1987.07.24

A labdarúgó kódja: EM23
A csapat kódja: 5
A labdarúgó posztja: Kapus
A labdarúgó keresztnéve: Emiliano
A labdarúgó vezetékneve: Martinez
Gólok száma: 0
Mezszám: 23
Születési dátum: 1992.09.02

--2.-Az-összes-csapat-neve--
Csapat neve: Portugália

Csapat neve: Németország

Csapat neve: Franciaország

Csapat neve: Anglia

Csapat neve: Argentína

--3.-Mérkőzés-dátuma-és-játékvezető-kiírása--

A mérkőzés dátuma: 2022.11.22

A játékvezető neve: Frank de Bleckere

A mérkőzés dátuma: 2022.11.26

A játékvezető neve: Pedro Proenca

A mérkőzés dátuma: 2022.11.30

A játékvezető neve: Kim Milton Nielsen

A mérkőzés dátuma: 2022.12.01

A játékvezető neve: Michel Vautrot

A mérkőzés dátuma: 2022.12.18

A játékvezető neve: Oscar Ruiz

--4.-'21'-es-ID-val-rendelkező-logó-tervezőjének-neve--

A '21'-es ID-val rendelkező logó tervezője: António Modesto

--5.-Az-összes-stadion-nevének-kiírása,-amelynek-a-férőhelyeinek-pontosan-40000-fő--

Stadion neve: al-Thumama Stadion

Stadion neve: El-Dzsanúb Stadion

5c.) DOM adatmódosítás

'DOMModifyDLWGQZ' osztály: Ez az osztály végzi el az XML fájlban a módosításokat a DOM használatával.

Adatmódosításhoz a következő segédmetódusokat használtam:

- **'getElementsByTagName':** Node-listát ad vissza, amely az adott elem nevű összes leszármazottját tartalmazza a hívott elemen belül.
- **'setTextContext':** beállítja az elem szövegtartalmát a megadott értékre.

'modifyXMLElements' metódus: ez a metódus fogja módosítani az 'XMLDLWGQZ.xml' fájlt.

Összesen 5 darab adatmódosítást hajtottam végre:

1. A német válogatott vezetőedzőjének módosítása
2. Ronaldo és Messi góljai számának módosítása
3. Education City Stadion, Al Bayt Stadion és az al-Thumama Stadion férőhelyeinek számainak módosítása
4. 2022.11.26-i mérkőzést dátumának módosítása
5. Csoport jelölésének módosítása

DOMModifyDLWGQZ.java forráskódja:

```
package hu.domparse.dlwqz;

import java.io.File;
import org.w3c.dom.*;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;

public class DOMModifyDLWGQZ {
    public static void main(String[] args) {
        // Metódus meghívása
        modifyXMLElements("XMLDLWGQZ.xml");
    }

    private static void modifyXMLElements(String filePath) {
        try {
            // Fájl beolvasása
            File xmlFile = new File(filePath);

            DocumentBuilderFactory dbFactory =
DocumentBuilderFactory.newInstance();
            // Ez a dokumentumépítő példányok létrehozására szolgál
            DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
            // Ez a dokumentum építésére szolgál
            Document document = dBuilder.parse(xmlFile);

            NodeList csapatList = document.getElementsByTagName("Csapat");
            // Lekérjük azt az elemet a listából amelyiket módosítani szeretnénk
            // index alapján történik a módosítás
            Element csapatElement = (Element) csapatList.item(1);

            // 1. módosítás
            // A német válogatott vezetőedző nevének módosítása
            csapatElement.getElementsByTagName("Vezetőedző").item(0).setTextContent("Julian Nagelsmann");

            // 2. módosítás
            // Ronaldo és Messi góljai számának módosítása
            NodeList labdarugoList = document.getElementsByTagName("Labdarúgó");
            Element labdarugoElement_CR7 = (Element) labdarugoList.item(0);
            labdarugoElement_CR7.getElementsByTagName("Gól").item(0).setTextContent("14");

            Element labdarugoElement_LM10 = (Element) labdarugoList.item(8);
```

```

        labdarugoElement_LM10.getElementsByTagName("Gól").item(0).setTextContent("13");

        // 3. módosítás
        // Education City Stadion, Al Bayt Stadion és az al-Thumama Stadion
        // férőhelyeinek számainak módosítása
        NodeList helyszinList = document.getElementsByTagName("Helyszín");
        Element helyszinElement_Education_City_Stadion = (Element)
helyszinList.item(0);
        helyszinElement_Education_City_Stadion.getElementsByTagName("Férőhely")
).item(0).setTextContent("41500");
        Element helyszinElement_Al_Bayt_Stadion = (Element)
helyszinList.item(1);
        helyszinElement_Al_Bayt_Stadion.getElementsByTagName("Férőhely").item(
0).setTextContent("55000");
        Element helyszinElement_al_Thumama_Stadion = (Element)
helyszinList.item(3);
        helyszinElement_al_Thumama_Stadion.getElementsByTagName("Férőhely").it
em(0).setTextContent("30000");

        // 4. módosítás
        // 2022.11.26-i mérkőzést dátumának módosítása
        NodeList merkozikList = document.getElementsByTagName("Mérkőzik");
        Element merkozikElement = (Element) merkozikList.item(1);
        merkozikElement.getElementsByTagName("Dátum").item(0).setTextContent("
2022.11.28");

        // 5. módosítás
        // Csoport jelölésének módosítása
        NodeList csoportList = document.getElementsByTagName("Csoport");
        Element csoportElement = (Element) csoportList.item(0);
        csoportElement.getElementsByTagName("Jelölés").item(0).setTextContent(
"G");

        // Kiírjuk a konzolra a módosított XML fájlt
        // A konzolra íratáshoz transformerFactoryt alkalmazom
        // Definiálok a factoryból egy új példányt
        TransformerFactory transformerFactory =
TransformerFactory.newInstance();

        // Beállítom a transformert
        Transformer transformer = transformerFactory.newTransformer();

        // Megadom a forrás fájlt
        DOMSource source = new DOMSource(document);

        // Megnyitom a streamet és konzolra kiíratom sys.out-al a fájlt
        StreamResult consoleResult = new StreamResult(System.out);

```

```

        transformer.transform(source, consoleResult);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

Output:

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?><Világbajnokság_DLWGQZ
xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
xs:noNamespaceSchemaLocation="XMLSchemaDLWGQZ.xsd">
  <!--Csapatok-->
  <Csapat CSOPkód="11" CSkód="1" Csapatkapitány="CR7" LOGÓkód="21">
    <Vezetőedző>Roberto Martínez</Vezetőedző>
    <Név>Portugália</Név>
  </Csapat>
  <Csapat CSOPkód="12" CSkód="2" Csapatkapitány="TS1" LOGÓkód="22">
    <Vezetőedző>Julian Nagelsmann</Vezetőedző>
    <Név>Németország</Név>
  </Csapat>
  <Csapat CSOPkód="13" CSkód="3" Csapatkapitány="KM10" LOGÓkód="23">
    <Vezetőedző>Didier Deschamp</Vezetőedző>
    <Név>Franciaország</Név>
  </Csapat>
  <Csapat CSOPkód="14" CSkód="4" Csapatkapitány="HK9" LOGÓkód="24">
    <Vezetőedző>Gareth Southgate</Vezetőedző>
    <Név>Anglia</Név>
  </Csapat>
  <Csapat CSOPkód="15" CSkód="5" Csapatkapitány="LM10" LOGÓkód="25">
    <Vezetőedző>Lionel Scaloni</Vezetőedző>
    <Név>Argentína</Név>
  </Csapat>
  <!--Logók-->
  <Logó LOGÓkód="21">
    <Színek>Piros</Színek>
    <Színek>Kék</Színek>
    <Színek>Zöld</Színek>
    <Tervező>António Modesto</Tervező>
  </Logó>
  <Logó LOGÓkód="22">
    <Színek>Fekete</Színek>
    <Színek>Piros</Színek>
    <Színek>Arany</Színek>
    <Tervező>Anton Stankowski</Tervező>
  </Logó>
  <Logó LOGÓkód="23">
    <Színek>Kék</Színek>
    <Színek>Fehér</Színek>
    <Színek>Piros</Színek>
    <Tervező>Raymond Savignac</Tervező>
  </Logó>
  <Logó LOGÓkód="24">
    <Színek>Fehér</Színek>

```

```

        <Színek>Kék</Színek>
        <Színek>Piros</Színek>
        <Tervező>William C. Gibbons</Tervező>
</Logó>
<Logó LOGÓkód="25">
        <Színek>Égkék</Színek>
        <Színek>Fehér</Színek>
        <Színek>Sárga</Színek>
        <Tervező>Salvador Dellutri</Tervező>
</Logó>
<!--Csoportok-->
<Csoport CSOPkód="11">
        <Jelölés>G</Jelölés>
</Csoport>
<Csoport CSOPkód="12">
        <Jelölés>B</Jelölés>
</Csoport>
<Csoport CSOPkód="13">
        <Jelölés>C</Jelölés>
</Csoport>
<Csoport CSOPkód="14">
        <Jelölés>D</Jelölés>
</Csoport>
<Csoport CSOPkód="15">
        <Jelölés>E</Jelölés>
</Csoport>
<!--Labdarúgók-->
<!--PORTUGÁL Labdarúgók-->
<Labdarúgó CSkód="1" Lkód="CR7">
        <Poszt>Középcsatár</Poszt>
        <Név>
                <Keresztnév>Cristiano</Keresztnév>
                <Vezetéknév>Ronaldo</Vezetéknév>
        </Név>
        <Gól>14</Gól>
        <Mezszám>7</Mezszám>
        <Szüldő>1985.02.05</Szüldő>
</Labdarúgó>
<Labdarúgó CSkód="1" Lkód="RN18">
        <Poszt>Középpályás</Poszt>
        <Poszt>Védekező középpályás</Poszt>
        <Név>
                <Keresztnév>Ruben</Keresztnév>
                <Vezetéknév>Neves</Vezetéknév>
        </Név>
        <Gól>2</Gól>
        <Mezszám>18</Mezszám>
        <Szüldő>1997.03.13</Szüldő>
</Labdarúgó>
<!--NÉMET Labdarúgók-->
<Labdarúgó CSkód="2" Lkód="TS1">
        <Poszt>Kapus</Poszt>
        <Név>
                <Keresztnév>Marc-André</Keresztnév>

```

```

        <Vezetéknév>ter Stegen</Vezetéknév>
    </Név>
    <Gól>0</Gól>
    <Mezszám>22</Mezszám>
    <Szüldő>1992.04.05</Szüldő>
</Labdarúgó>
<Labdarúgó CSKód="2" Lkód="JK6">
    <Poszt>Középpályás</Poszt>
    <Poszt>Jobb szélső védő</Poszt>
    <Név>
        <Keresztnév>Joshua</Keresztnév>
        <Vezetéknév>Kimmich</Vezetéknév>
    </Név>
    <Gól>3</Gól>
    <Mezszám>6</Mezszám>
    <Szüldő>1995.02.08</Szüldő>
</Labdarúgó>
<!-- FRANCIA Labdarúgók -->
<Labdarúgó CSKód="3" Lkód="KM10">
    <Poszt>Középcsatár</Poszt>
    <Név>
        <Keresztnév>Kylian</Keresztnév>
        <Vezetéknév>Mbappé</Vezetéknév>
    </Név>
    <Gól>9</Gól>
    <Mezszám>10</Mezszám>
    <Szüldő>1998.12.20</Szüldő>
</Labdarúgó>
<Labdarúgó CSKód="3" Lkód="EC25">
    <Poszt>Középpályás</Poszt>
    <Poszt>Védekező középpályás</Poszt>
    <Poszt>Bal szélső védő</Poszt>
    <Név>
        <Keresztnév>Eduardo</Keresztnév>
        <Vezetéknév>Camavinga</Vezetéknév>
    </Név>
    <Gól>3</Gól>
    <Mezszám>25</Mezszám>
    <Szüldő>2002.11.10</Szüldő>
</Labdarúgó>
<!-- ANGOL Labdarúgók -->
<Labdarúgó CSKód="4" Lkód="HK9">
    <Poszt>Középcsatár</Poszt>
    <Név>
        <Keresztnév>Harry</Keresztnév>
        <Vezetéknév>Kane</Vezetéknév>
    </Név>
    <Gól>7</Gól>
    <Mezszám>9</Mezszám>
    <Szüldő>1993.07.28</Szüldő>
</Labdarúgó>
<Labdarúgó CSKód="4" Lkód="JB10">
    <Poszt>Középpályás</Poszt>
    <Név>

```

```

        <Keresztnév>Jude</Keresztnév>
        <Vezetéknév>Bellingham</Vezetéknév>
    </Név>
    <Gól>6</Gól>
    <Mezszám>10</Mezszám>
    <Szüldő>2003.06.29</Szüldő>
</Labdarúgó>
<!--ARGENTÍN Labdarúgók-->
<Labdarúgó CSkód="5" Lkód="LM10">
    <Poszt>Középcsatár</Poszt>
    <Poszt>Jobb szélső csatár</Poszt>
    <Név>
        <Keresztnév>Lionel</Keresztnév>
        <Vezetéknév>Messi</Vezetéknév>
    </Név>
    <Gól>13</Gól>
    <Mezszám>10</Mezszám>
    <Szüldő>1987.07.24</Szüldő>
</Labdarúgó>
<Labdarúgó CSkód="5" Lkód="EM23">
    <Poszt>Kapus</Poszt>
    <Név>
        <Keresztnév>Emiliano</Keresztnév>
        <Vezetéknév>Martinez</Vezetéknév>
    </Név>
    <Gól>0</Gól>
    <Mezszám>23</Mezszám>
    <Szüldő>1992.09.02</Szüldő>
</Labdarúgó>
<!--Mérkőzések-->
<Mérkőzés Hkód="201" Mkód="1" Tkód="101">
    <Csapat1Gólok>5</Csapat1Gólok>
    <Csapat2Gólok>2</Csapat2Gólok>
</Mérkőzés>
<Mérkőzés Hkód="202" Mkód="2" Tkód="102">
    <Csapat1Gólok>0</Csapat1Gólok>
    <Csapat2Gólok>1</Csapat2Gólok>
</Mérkőzés>
<Mérkőzés Hkód="203" Mkód="3" Tkód="103">
    <Csapat1Gólok>3</Csapat1Gólok>
    <Csapat2Gólok>7</Csapat2Gólok>
</Mérkőzés>
<Mérkőzés Hkód="204" Mkód="4" Tkód="104">
    <Csapat1Gólok>2</Csapat1Gólok>
    <Csapat2Gólok>1</Csapat2Gólok>
</Mérkőzés>
<Mérkőzés Hkód="205" Mkód="5" Tkód="105">
    <Csapat1Gólok>2</Csapat1Gólok>
    <Csapat2Gólok>3</Csapat2Gólok>
</Mérkőzés>
<!--Helyszínek-->
<Helyszín Hkód="201">
    <Stadion>
        <Név>Education City Stadion</Név>

```

```

        <Férőhely>41500</Férőhely>
        <Tulajdonos>katari kormány</Tulajdonos>
    </Stadion>
    <Város>al-Rajján</Város>
</Helyszín>
<Helyszín Hkód="202">
    <Stadion>
        <Név>Al Bayt Stadion</Név>
        <Férőhely>55000</Férőhely>
        <Tulajdonos>katari kormány</Tulajdonos>
    </Stadion>
    <Város>al-Hor</Város>
</Helyszín>
<Helyszín Hkód="203">
    <Stadion>
        <Név>Loszaíli Nemzeti Stadion</Név>
        <Férőhely>80000</Férőhely>
        <Tulajdonos>katari kormány</Tulajdonos>
    </Stadion>
    <Város>Loszaíl</Város>
</Helyszín>
<Helyszín Hkód="204">
    <Stadion>
        <Név>al-Thumama Stadion</Név>
        <Férőhely>30000</Férőhely>
        <Tulajdonos>katari kormány</Tulajdonos>
    </Stadion>
    <Város>al-Thumama</Város>
</Helyszín>
<Helyszín Hkód="205">
    <Stadion>
        <Név>El-Dzsanúb Stadion</Név>
        <Férőhely>40000</Férőhely>
        <Tulajdonos>katari kormány</Tulajdonos>
    </Stadion>
    <Város>Al-Vakra</Város>
</Helyszín>
<!--Típusok-->
<Típus Tkód="101">
    <Név>Csoportmérkőzés</Név>
</Típus>
<Típus Tkód="102">
    <Név>Nyolcaddöntő</Név>
</Típus>
<Típus Tkód="103">
    <Név>Negyeddöntő</Név>
</Típus>
<Típus Tkód="104">
    <Név>Elődöntő</Név>
</Típus>
<Típus Tkód="105">
    <Név>Döntő</Név>
</Típus>
<!--Mérkőzik kapcsolat-->

```



```

<Mérkőzik Csapat1="1" Csapat2="2" Győztes="1" Mkód="1">
  <Dátum>2022.11.22</Dátum>
  <Játékvezető>Frank de Bleeckere</Játékvezető>
</Mérkőzik>
<Mérkőzik Csapat1="3" Csapat2="5" Győztes="5" Mkód="2">
  <Dátum>2022.11.28</Dátum>
  <Játékvezető>Pedro Proenca</Játékvezető>
</Mérkőzik>
<Mérkőzik Csapat1="2" Csapat2="1" Győztes="2" Mkód="3">
  <Dátum>2022.11.30</Dátum>
  <Játékvezető>Kim Milton Nielsen</Játékvezető>
</Mérkőzik>
<Mérkőzik Csapat1="4" Csapat2="5" Győztes="4" Mkód="4">
  <Dátum>2022.12.01</Dátum>
  <Játékvezető>Michel Vautrot</Játékvezető>
</Mérkőzik>
<Mérkőzik Csapat1="1" Csapat2="5" Győztes="1" Mkód="5">
  <Dátum>2022.12.18</Dátum>
  <Játékvezető>Oscar Ruiz</Játékvezető>
</Mérkőzik>
</Világbajnokság_DLWGQZ>

```

5d.) DOM adatírás

'DOMWriteDLWGQZ' osztály: Ez az osztály valósítja meg az XML dokumentum létrehozását és abban való írását.

'writeElementsToFileAndConsole' metódus: előkészíti a dokumentumot, hozzáad különböző elemeket, majd kiírja a dokumentumot a konzolra és egy XML fájlba eltárolja.

Az elemek hozzáadásához metódusokat készítettem:

- 'addCsapat'
- 'addLogo'
- 'addCsoport'
- 'addLabdarugo'
- 'addMerkozes'
- 'addHelyszin'
- 'addTipus'
- 'addMerkozik'

'printNodeList' metódus: Segítségével kiírja a NodeList elemeit a konzolra és a PrintWriter-re.

'printNode' metódus: Segítségével kiírja az adott Node-ot a konzolra és a PrintWriter-re.

'printDocument' metódus: Segítségével kiírja a dokumentumot a konzolra és a PrintWriter-re.

'getIndentString' metódus: Ez a metódus a behúzás mértékétől függően előállít egy **String**-et, amely tartalmazza a megfelelő számú szóközt.

DOMWriteDLWGQZ.java forráskódja:

```
package hu.domparse.dlwqz;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.transform.OutputKeys;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import java.io.File;
import java.io.FileWriter;
import java.io.PrintWriter;
import java.util.Arrays;
import java.util.List;
import java.util.StringJoiner;

import org.w3c.dom.*;

public class DOMWriteDLWGQZ {
    public static void main(String[] args) {
        // Metódus meghívása
        writeElementsToFileAndConsole();
    }

    private static void writeElementsToFileAndConsole() {
        try {
            // Előkészítjük a dokumentumot
            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
            // Ez a dokumentumépítő példányok létrehozására szolgál
            DocumentBuilder builder = factory.newDocumentBuilder();
            // Ez a dokumentum építésére szolgál
            Document document = builder.newDocument();
            // Gyökérelem létrehozása
            Element rootElement = document.createElement("Világbajnokság_DLWGQZ");
            rootElement.setAttribute("xmlns:xsi",
                "http://www.w3.org/2001/XMLSchema-instance");
            rootElement.setAttribute("xsi:noNamespaceSchemaLocation",
                "XMLSchemaDLWGQZ.xsd");
            document.appendChild(rootElement);

            // Csapat elemek létrehozása
            addCsapat(document, rootElement, "1", "CR7", "11", "21", "Roberto
Martínez", "Portugália");
            addCsapat(document, rootElement, "2", "TS1", "12", "22", "Hansi
Flick", "Németország");
            addCsapat(document, rootElement, "3", "KM10", "13", "23", "Didier
Deschamp", "Franciaország");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```

        addCsapat(document, rootElement, "4", "HK9", "14", "24", "Gareth
Southgate", "Anglia");
        addCsapat(document, rootElement, "5", "LM10", "15", "25", "Lionel
Scaloni", "Argentína");

        // Logó elemek létrehozása
        addLogo(document, rootElement, "21", Arrays.asList("Piros", "Kék",
"Zöld"), "António Modesto");
        addLogo(document, rootElement, "22", Arrays.asList("Fekete", "Piros",
"Arany"), "Anton Stankowski");
        addLogo(document, rootElement, "23", Arrays.asList("Kék", "Fehér",
"Piros"), "Raymond Savignac");
        addLogo(document, rootElement, "24", Arrays.asList("Fehér", "Kék",
"Piros"), "William C. Gibbons");
        addLogo(document, rootElement, "25", Arrays.asList("Égkék", "Fehér",
"Sárga"), "Salvador Dellutri");

        // Csoport elemek létrehozása
        addCsoport(document, rootElement, "11", "A");
        addCsoport(document, rootElement, "12", "B");
        addCsoport(document, rootElement, "13", "C");
        addCsoport(document, rootElement, "14", "D");
        addCsoport(document, rootElement, "15", "E");

        // Labdarúgó elemek létrehozása
        addLabdarugo(document, rootElement, "CR7", "1",
Arrays.asList("Középcsatár"), "Cristiano", "Ronaldo", "10", "7", "1985.02.05");
        addLabdarugo(document, rootElement, "RN18", "1",
Arrays.asList("Középpályás", "Védekező középpályás"), "Ruben", "Neves", "2", "18",
"1997.03.13");
        addLabdarugo(document, rootElement, "TS1", "2",
Arrays.asList("Kapus"), "Marc-André", "ter Stegen", "0", "22", "1992.04.05");
        addLabdarugo(document, rootElement, "JK6", "2",
Arrays.asList("Középpályás", "Jobb szélső védő"), "Joshua", "Kimmich", "3", "6",
"1995.02.08");
        addLabdarugo(document, rootElement, "KM10", "3",
Arrays.asList("Középcsatár"), "Kylia", "Mbappé", "9", "10", "1998.12.20");
        addLabdarugo(document, rootElement, "EC25", "3",
Arrays.asList("Középpályás", "Védekező középpályás", "Bal szélső védő"),
"Eduardo", "Camavinga", "3", "25", "2002.11.10");
        addLabdarugo(document, rootElement, "HK9", "4",
Arrays.asList("Középcsatár"), "Harry", "Kane", "7", "9", "1993.07.28");
        addLabdarugo(document, rootElement, "JB10", "4",
Arrays.asList("Középpályás"), "Jude", "Bellingham", "6", "10", "2003.06.29");
        addLabdarugo(document, rootElement, "LM10", "5",
Arrays.asList("Középcsatár", "Jobb szélső csatár"), "Lionel", "Messi", "9", "10",
"1987.07.24");

```

```

        addLabdarugo(document, rootElement, "EM23", "5",
Arrays.asList("Kapus"), "Emiliano", "Martinez", "0", "23", "1992.09.02");

// Mérkőzés elemek létrehozása
addMerkozes(document, rootElement, "1", "101", "201", "5", "2");
addMerkozes(document, rootElement, "2", "102", "202", "0", "1");
addMerkozes(document, rootElement, "3", "103", "203", "3", "7");
addMerkozes(document, rootElement, "4", "104", "204", "2", "1");
addMerkozes(document, rootElement, "5", "104", "205", "2", "3");

// Helyszín elemek létrehozása
addHelyszin(document, rootElement, "201", "Education City Stadion",
"45000", "katari kormány", "al-Rajján");
addHelyszin(document, rootElement, "202", "Al Bayt Stadion", "60000",
"katari kormány", "al-Hor");
addHelyszin(document, rootElement, "203", "Loszaíli Nemzeti Stadion",
"80000", "katari kormány", "Loszaíl");
addHelyszin(document, rootElement, "204", "al-Thumama Stadion",
"40000", "katari kormány", "al-Thumama");
addHelyszin(document, rootElement, "205", "El-Dzsanúb Stadion",
"40000", "katari kormány", "Al-Vakra");

// Típus elemek létrehozása
addTipus(document, rootElement, "101", "Csoportmérkőzés");
addTipus(document, rootElement, "102", "Nyolcaddöntő");
addTipus(document, rootElement, "103", "Negyedöntő");
addTipus(document, rootElement, "104", "Elődöntő");
addTipus(document, rootElement, "105", "Döntő");

// Mérkőzik elemek létrehozása
addMerkozik(document, rootElement, "1", "1", "2", "1", "2022.11.22",
"Frank de Bleeckere");
addMerkozik(document, rootElement, "2", "3", "5", "5", "2022.11.26",
"Pedro Proenca");
addMerkozik(document, rootElement, "3", "2", "1", "2", "2022.11.30",
"Kim Milton Nielsen");
addMerkozik(document, rootElement, "4", "4", "5", "4", "2022.12.01",
"Michel Vautrot");
addMerkozik(document, rootElement, "5", "1", "5", "1", "2022.12.18",
"Oscar Ruiz");

// Dokumentum mentése
TransformerFactory transformerFactory =
TransformerFactory.newInstance();
Transformer transformer = transformerFactory.newTransformer();
transformer.setOutputProperty(OutputKeys.ENCODING, "UTF-8");
transformer.setOutputProperty(OutputKeys.INDENT, "yes");

```

```

        transformer.setOutputProperty("{https://xml.apache.org/xslt}indent-amount", "4");

        printDocument(document);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

//addMerkozik metódus (MÉRKŐZIK)
private static void addMerkozik(Document document, Element rootElement, String
mkod, String csapat1, String csapat2,
    String gyoztes, String datum, String jatekvezeto) {
    Element merkozik = document.createElement("Mérkőzik");
    merkozik.setAttribute("Mkód", mkod);
    merkozik.setAttribute("Csapat1", csapat1);
    merkozik.setAttribute("Csapat2", csapat2);
    merkozik.setAttribute("Győztes", gyoztes);

    Element datumElement = createElement(document, "Dátum", datum);
    merkozik.appendChild(datumElement);

    Element jatekvezetoElement = createElement(document, "Játékvezető",
jatekvezeto);
    merkozik.appendChild(jatekvezetoElement);

    rootElement.appendChild(merkozik);
}

//addTipus metódus (TÍPUS)
private static void addTipus(Document document, Element rootElement, String
tkod, String nev) {
    Element tipus = document.createElement("Típus");
    tipus.setAttribute("Tkód", tkod);

    Element nevElement = createElement(document, "Név", nev);
    tipus.appendChild(nevElement);

    rootElement.appendChild(tipus);
}

//addHelyszin metódus (HELYSZÍN)
private static void addHelyszin(Document document, Element rootElement, String
hkod, String nev, String ferohely,
    String tulajdonos, String varos) {
    Element helyszin = document.createElement("Helyszín");
    helyszin.setAttribute("Hkód", hkod);

```

```

        Element stadionElement = document.createElement("Stadion");
        Element nevElement = createElement(document, "Név", nev);
        Element ferohelyElement = createElement(document, "Férőhely", ferohely);
        Element tulajdonosElement = createElement(document, "Tulajdonos",
tulajdonos);
        stadionElement.appendChild(nevElement);
        stadionElement.appendChild(ferohelyElement);
        stadionElement.appendChild(tulajdonosElement);
        helyszin.appendChild(stadionElement);

        Element varosElement = createElement(document, "Város", varos);
        helyszin.appendChild(varosElement);

        rootElement.appendChild(helyszin);
    }

    //addMerkozes metódus (MÉRKŐZÉS)
    private static void addMerkozes(Document document, Element rootElement, String
mkod, String tkod, String hkod,
        String csapat1golo, String csapat2golo) {
        Element merkozes = document.createElement("Mérkőzés");
        merkozes.setAttribute("Mkód", mkod);
        merkozes.setAttribute("Tkód", tkod);
        merkozes.setAttribute("Hkód", hkod);

        Element csapat1goloElement = createElement(document, "Csapat1Gól",
csapat1golo);
        merkozes.appendChild(csapat1goloElement);

        Element csapat2goloElement = createElement(document, "Csapat2Gól",
csapat2golo);
        merkozes.appendChild(csapat2goloElement);

        rootElement.appendChild(merkozes);
    }

    //addLabdarugo metódus (LABDARÚGÓ)
    private static void addLabdarugo(Document document, Element rootElement,
String lkod, String cskod, List<String> posztok,
        String keresztnev, String vezeteknev, String gol, String mezzsam,
String szulido) {
        Element labdarugo = document.createElement("Labdarúgó");
        labdarugo.setAttribute("Lkód", lkod);
        labdarugo.setAttribute("CSkód", cskod);

        for (String poszt : posztok) {
            Element posztElement = createElement(document, "Poszt", poszt);

```

```

        labdarugo.appendChild(posztElement);
    }

    Element nevElement = document.createElement("Név");
    Element keresztnevElement = createElement(document, "Keresztnév",
keresztnev);
    Element vezeteknevElement = createElement(document, "Vezetéknév",
vezeteknev);
    nevElement.appendChild(keresztnevElement);
    nevElement.appendChild(vezeteknevElement);
    labdarugo.appendChild(nevElement);

    Element golElement = createElement(document, "Gól", gol);
    labdarugo.appendChild(golElement);

    Element mezzszámElement = createElement(document, "Mezszám", mezzszám);
    labdarugo.appendChild(mezzszámElement);

    Element szulidoElement = createElement(document, "Szüldő", szulido);
    labdarugo.appendChild(szulidoElement);

    rootElement.appendChild(labdarugo);
}

//addCsoport metódus (CSOPORT)
private static void addCsoport(Document document, Element rootElement, String
csopkod, String jeloles) {
    Element csoport = document.createElement("Csoport");
    csoport.setAttribute("CSOPkód", csopkod);

    Element jelolesElement = createElement(document, "Jelölés", jeloles);
    csoport.appendChild(jelolesElement);

    rootElement.appendChild(csoport);
}

//addLogo metódus (LOGÓ)
private static void addLogo(Document document, Element rootElement, String
logokod, List<String> szinek, String tervezo) {
    Element logo = document.createElement("Logó");
    logo.setAttribute("LOGÓKód", logokod);

    for (String szin : szinek) {
        Element szinElement = createElement(document, "Színek", szin);
        logo.appendChild(szinElement);
    }
}

```

```

        Element tervezoElement = createElement(document, "Tervező", tervezo);
        logo.appendChild(tervezoElement);

        rootElement.appendChild(logo);
    }

    //addCsapat metódus (CSAPAT)
    private static void addCsapat(Document document, Element rootElement, String
cskod, String csapatkapitany, String csopkod,
        String logokod, String vezetoedzo, String nev ) {
        Element csapat = document.createElement("Csapat");
        csapat.setAttribute("CSkód", cskod);
        csapat.setAttribute("Csapatkapitány", csapatkapitany);
        csapat.setAttribute("CSOPkód", csopkod);
        csapat.setAttribute("LOGÓkód", logokod);

        Element vezetoedzoElement = createElement(document, "Vezetőedző",
vezetoedzo);
        Element nevElement = createElement(document, "Név", nev);
        csapat.appendChild(vezetoedzoElement);
        csapat.appendChild(nevElement);

        rootElement.appendChild(csapat);
    }

    private static Element createElement(Document document, String name, String
value) {
        Element element = document.createElement(name);
        element.appendChild(document.createTextNode(value));
        return element;
    }

    private static void printNodeList(NodeList nodeList, PrintWriter writer) {
        for (int i = 0; i < nodeList.getLength(); i++) {
            Node node = nodeList.item(i);
            printNode(node, 1, writer);
            System.out.println("");
            writer.println("");
        }
    }

    private static void printNode(Node node, int indent, PrintWriter writer) {
        // Ha az elem típusa ELEMENT_NODE, akkor kiírjuk az elem nevét és
attribútumait
        if (node.getNodeType() == Node.ELEMENT_NODE) {
            Element element = (Element) node;

```



```

        String nodeName = element.getTagName();
        StringJoiner attributes = new StringJoiner(" ");
        NamedNodeMap attributeMap = element.getAttributes();
        // Kiírjuk az elem nevét és attribútumait
        for (int i = 0; i < attributeMap.getLength(); i++) {
            Node attribute = attributeMap.item(i);
            attributes.add(attribute.getNodeName() + "=\"" +
attribute.getNodeValue() + "\"");
        }

        // Kiírjuk az elem nevét és attribútumait
        System.out.print(getIndentString(indent));
        System.out.print("<" + nodeName + " " + attributes.toString() +
">");

        writer.print(getIndentString(indent));
        writer.print("<" + nodeName + " " + attributes.toString() + ">");

        NodeList children = element.getChildNodes();
        if (children.getLength() == 1 && children.item(0).getNodeType() ==
Node.TEXT_NODE) {
            System.out.print(children.item(0).getNodeValue());
            writer.print(children.item(0).getNodeValue());
        } else {
            System.out.println();
            writer.println();
            for (int i = 0; i < children.getLength(); i++) {
                printNode(children.item(i), indent + 1, writer);
            }
            System.out.print(getIndentString(indent));
            writer.print(getIndentString(indent));
        }
        System.out.println("</" + nodeName + ">");
        writer.println("</" + nodeName + ">");
    }
}

private static void printDocument(Document document) {
    try {
        // Fájlba írás
        File xmlFile = new File("XMLDLWGQZ1.xml");

        // Írás a konzolra
        PrintWriter writer = new PrintWriter(new FileWriter(xmlFile,
true));

        // Kiírja az XML főgyökér elemét a konzolra és fájlba
        Element rootElement = document.getDocumentElement();

```

```

String rootName = rootElement.getTagName();

// A gyökérelem attribútumainak kiírása
StringJoiner rootAttributes = new StringJoiner(" ");

// Gyökérelem attribútumainak lekérése
NamedNodeMap rootAttributeMap = rootElement.getAttributes();

for (int i = 0; i < rootAttributeMap.getLength(); i++) {
    Node attribute = rootAttributeMap.item(i);
    rootAttributes.add(attribute.getNodeName() + "=\"" +
attribute.getNodeValue() + "\"");
}

System.out.print("<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n");
writer.print("<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n");

System.out.print("<" + rootName + " " + rootAttributes.toString()
+ ">\n");
writer.print("<" + rootName + " " + rootAttributes.toString() +
">\n");

// A gyökér elem alatti elemek lekérése
NodeList csapatList = document.getElementsByTagName("Csapat");
NodeList logoList = document.getElementsByTagName("Logó");
NodeList csoportList = document.getElementsByTagName("Csoport");
NodeList labdarugóList =
document.getElementsByTagName("Labdarúgó");
NodeList merkozesList = document.getElementsByTagName("Mérkőzés");
NodeList helyszinList = document.getElementsByTagName("Helyszín");
NodeList tipusList = document.getElementsByTagName("Típus");
NodeList merkozikList = document.getElementsByTagName("Mérkőzik");

printNodeList(csapatList, writer);
System.out.println("");
writer.println("");

printNodeList(logoList, writer);
System.out.println("");
writer.println("");

printNodeList(csoportList, writer);
System.out.println("");
writer.println("");

printNodeList(labdarugóList, writer);
System.out.println("");
writer.println("");

```

```

        printNodeList(merkozsesList, writer);
        System.out.println("");
        writer.println("");

        printNodeList(helyszinList, writer);
        System.out.println("");
        writer.println("");

        printNodeList(tipusList, writer);
        System.out.println("");
        writer.println("");

        printNodeList(merkozikList, writer);
        System.out.println("");
        writer.println("");

        System.out.println("</" + rootName + ">");
        writer.append("</" + rootName + ">");

        writer.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

private static String getIndentString(int indent) {
    StringBuilder sb = new StringBuilder();
    for (int i = 0; i < indent; i++) {
        sb.append(" ");
    }
    return sb.toString();
}
}

```