

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Táncverseny

Készítette: Baffi Réka

Neptunkód: F4IIYA

Dátum: 2023.12.05

Tartalom

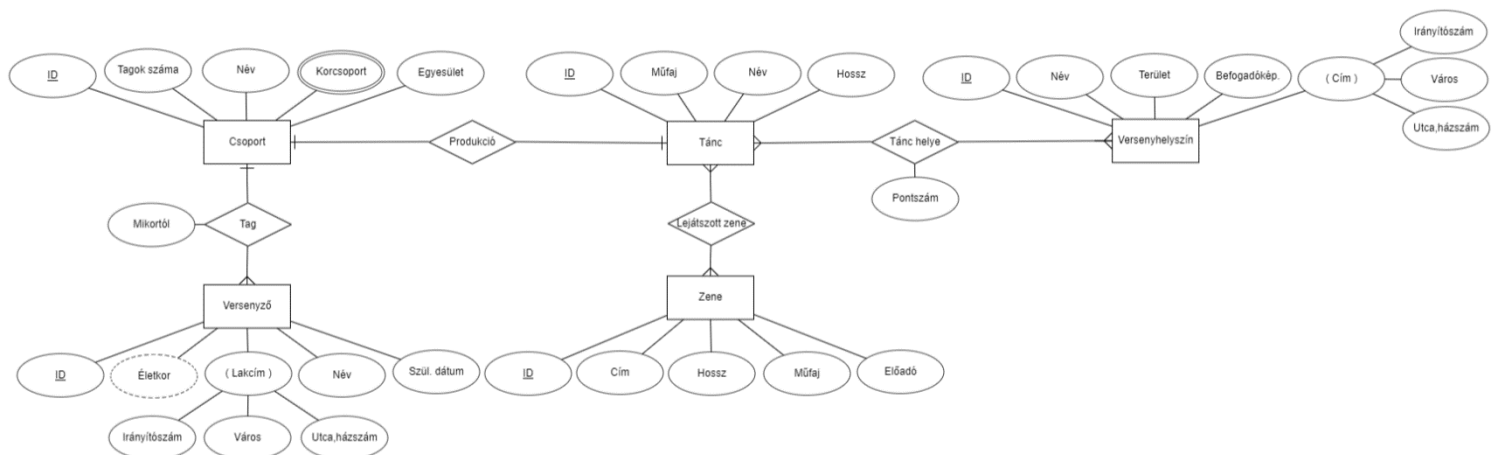
1.	Feladat témája	3
2.	Az ER modell konvertálása XDM modellé.....	4
3.	XML dokumentum készítése.....	5
4.	XMLSchema készítése	12
5.	DOM adatolvasás	17
6.	DOM adatmódosítás	24
7.	DOM adatlekérdezés.....	26
8.	DOM adatírás	33

1. Feladat témája

A beadandó témája egy olyan adatbázis, amely több helyszínen megrendezendő táncversenyek különböző adatait számon tartja. Lehetőség van egyes versenyzők, illetve csoportok, vagy akár az adott produkciók adatainak lekérdezésére.

- Versenyhelyszín egyed tulajdonságai
 - o id: a versenyhelyszín kulcsa
 - o név: a helyszín neve
 - o terület: mekkora területen helyezkedik el
 - o befogadóképesség: hány főt tud befogadni
 - o cím: összetett tulajdonság, mely a város, irányítószám és utca, házszámból áll
- Tánc egyed tulajdonságai
 - o id: a tánc egyedi kulcsa
 - o név: a produkció neve
 - o hossz: a produkció hossza, percben
 - o műfaj: a tánc műfaja
- Zene egyed tulajdonságai
 - o id: a zene egyedi kulcsa
 - o cím: a zene címe
 - o hossz: a zene hossza
 - o műfaj: a zene műfaja
 - o előadó: a zene előadója
- Csoport egyed tulajdonságai
 - o id: csoport egyedi azonosítója
 - o tagok száma: hányan vannak a csoportban
 - o név: mi a csoport neve
 - o egyesület: melyik egyesülethez tartozik a csoport
 - o korcsoport: korcsoport, amelyben az adott csoport versenyezik, ez lehet többértékű
- Versenyző egyed tulajdonságai
 - o id: versenyző kulcsa
 - o név: a versenyző neve
 - o szül. dátum: versenyző születési dátuma
 - o lakcím: összetett tulajdonság, mely a város, irányítószám és utca, házszámból áll
 - o életkor: opcionális mező, amelyben meg lehet adni a versenyző életkorát

A feladat ER modellje:



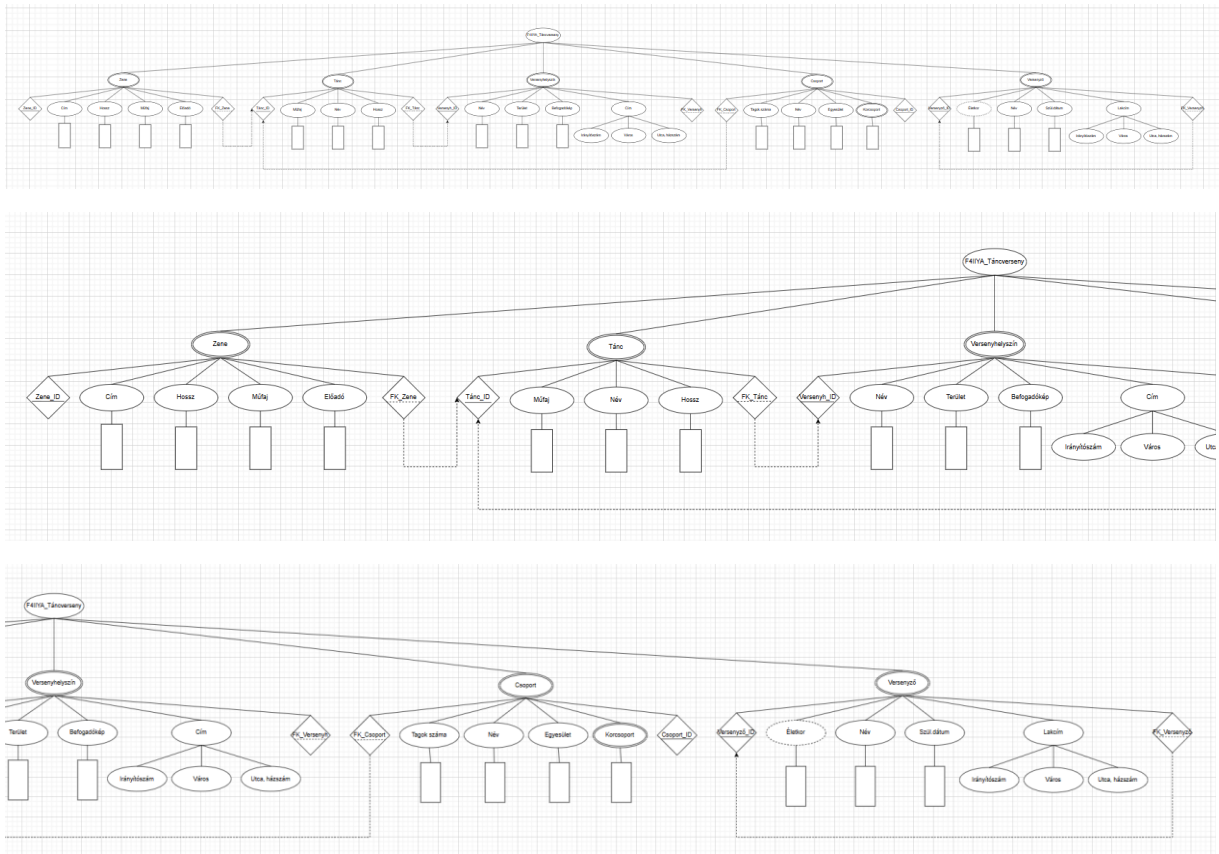
Az egyedek közötti kapcsolatok:

- Csoport és Versenyző kapcsolata: Tag
 - o Egy-több kapcsolat van közöttük, mivel egy csoporthoz több versenyző is tartozhat.
- Csoport és Tánc kapcsolata: Produkció
 - o Egy-egy kapcsolat van közöttük, mivel egy versenyen egy produkciót tudnak előadni.
- Tánc és Zene kapcsolata: Lejátszott zene
 - o Több-több kapcsolat van közöttük, mivel ugyanazt a zenét több tánc is felhasználhatja, illetve egy tánc előadás alatt több zenét is fel lehet használni.
- Tánc és Versenyhelyszín kapcsolata: Tánc helye
 - o Több-több kapcsolat jelenik meg itt is, mivel egy táncot több helyszínen is elő lehet adni, több verseny keretein belül, illetve egy helyszínen, mivel versenyről van szó, több tánc is van előadva.

2. Az ER modell konvertálása XDM modellé

Az XDM modell alkalmazása során három különböző jelölést használunk: ellipszis, rombusz és téglalap. Az ellipszis az elemeket szimbolizálja, minden egyedi entitás elemmé válik, beleértve a tulajdonságokat is. A rombusz az attribútumokat ábrázolja, melyek a kulcs tulajdonságokból származnak. A téglalap a szöveget jelöli, amely később megjelenik az XML dokumentumban. Ha egy elem többször is előfordulhat, akkor a jelölése dupla ellipszissel történik. Az idegenkulcsok és a kulcsok közötti kapcsolatot szaggatott vonalas nyíllal ábrázoljuk.

A feladat XDM modellje:



3. XML dokumentum készítése

Az XML dokumentumot az XDM modell alapján készítettem el, kezdetben létrehozva a gyökérelémet, amely a " F4IIYA_Tancverseny" volt. Ezt követően minden gyökérelémhez legalább három példányt hoztam létre. Az egyes elemek attribútumaiban megtalálhatók kulcsok és idegenkulcsok is. Emellett a szülőelemeknek is létrehoztam az őket alkotó elemeket. A többértékű elemek esetén több példányt is létrehoztam, és előfordult, hogy ezeknek az elemeknek is voltak saját gyerekelemeik.

XML dokumentum forráskódja

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

```
<F4IIYA_Tancverseny xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
xs:noNamespaceSchemaLocation="XMLSchemaF4IIYA.xsd">
```

```
<!--Zenék-->
```

```
<zene zene_id="1">
  <cim>Ébredj fel</cim>
  <hossz>3:45</hossz>
  <mufaj>Pop</mufaj>
  <eloado>Kovács Anna</eloado>
</zene>
```

```
<zene zene_id="2">
  <cim>Korallpart</cim>
  <hossz>4:20</hossz>
  <mufaj>Rock</mufaj>
  <eloado>Szabó Bence és a Hangok</eloado>
</zene>

<zene zene_id="3">
  <cim>Táncolj!</cim>
  <hossz>2:58</hossz>
  <mufaj>Elektronikus</mufaj>
  <eloado>Electro Vibes</eloado>
</zene>

<zene zene_id="4">
  <cim>Szerelmi történet</cim>
  <hossz>5:12</hossz>
  <mufaj>Romantikus</mufaj>
  <eloado>Nagy Eszter és a Szívhangok</eloado>
</zene>

<zene zene_id="5">
  <cim>Éjszakai láz</cim>
  <hossz>3:30</hossz>
  <mufaj>Disco</mufaj>
  <eloado>DiscoFever Band</eloado>
</zene>

<!--Versenyhelyszínek-->

<versenyhelyszin verseny_id="1">
  <nev>Park Stadion</nev>
  <terulet>800 négyzetméter</terulet>
  <befogadokepesseg>5000</befogadokepesseg>
  <cim>
    <iranyitoszam>1024</iranyitoszam>
    <varos>Budapest</varos>
    <utca_hazszam>Kossuth utca 10</utca_hazszam>
  </cim>
</versenyhelyszin>

<versenyhelyszin verseny_id="2">
  <nev>Sportaréna</nev>
  <terulet>1200 négyzetméter</terulet>
  <befogadokepesseg>10000</befogadokepesseg>
  <cim>
    <iranyitoszam>1132</iranyitoszam>
    <varos>Budapest</varos>
```

```
        <utca_hazszam>Rákospatak utca 5</utca_hazszam>
    </cim>
</versenyhelyszin>
```

```
<versenyhelyszin verseny_id="3">
    <nev>Sportközpont</nev>
    <terulet>600 négyzetméter</terulet>
    <befogadokepesseg>3000</befogadokepesseg>
    <cim>
        <iranyitoszam>3521</iranyitoszam>
        <varos>Debrecen</varos>
        <utca_hazszam>Sport utca 8</utca_hazszam>
    </cim>
</versenyhelyszin>
```

```
<versenyhelyszin verseny_id="4">
    <nev>Városi Stadion</nev>
    <terulet>1000 négyzetméter</terulet>
    <befogadokepesseg>7000</befogadokepesseg>
    <cim>
        <iranyitoszam>4023</iranyitoszam>
        <varos>Debrecen</varos>
        <utca_hazszam>Sport tér 15</utca_hazszam>
    </cim>
</versenyhelyszin>
```

```
<versenyhelyszin verseny_id="5">
    <nev>Park Stadion</nev>
    <terulet>850 négyzetméter</terulet>
    <befogadokepesseg>5500</befogadokepesseg>
    <cim>
        <iranyitoszam>1015</iranyitoszam>
        <varos>Budapest</varos>
        <utca_hazszam>Sport út 20</utca_hazszam>
    </cim>
</versenyhelyszin>
```

```
<!--Táncok-->
```

```
<tanc tanc_id="1">
    <t_mufaj>Modern tánc</t_mufaj>
    <nev>Fények játéka</nev>
    <hossz>4:30</hossz>
</tanc>
```

```
<tanc tanc_id="2">
    <t_mufaj>Latin tánc</t_mufaj>
    <nev>Forró ritmusok</nev>
    <hossz>3:15</hossz>
```

</tanc>

<tanc tanc_id="3">
 <t_mufaj>Balett</t_mufaj>
 <nev>Hamupipőke álom</nev>
 <hossz>5:00</hossz>
</tanc>

<tanc tanc_id="4">
 <t_mufaj>Jazz tánc</t_mufaj>
 <nev>Éjszakai szimfónia</nev>
 <hossz>3:45</hossz>
</tanc>

<tanc tanc_id="5">
 <t_mufaj>Flamenco</t_mufaj>
 <nev>Tűz és szenvedély</nev>
 <hossz>4:20</hossz>
</tanc>

<!--Csoportok-->

<csoport csoport_id="1" produkcio="1">
 <tagok_szama>15</tagok_szama>
 <nev>Arany Csillagok</nev>
 <egyesulet>Táncművészeti Egyesület</egyesulet>
 <korcsoport>25-30</korcsoport>
</csoport>

<csoport csoport_id="2" produkcio="2">
 <tagok_szama>10</tagok_szama>
 <nev>Kis Angyalok</nev>
 <egyesulet>Balett Stúdió</egyesulet>
 <korcsoport>12-18</korcsoport>
 <korcsoport>12-14</korcsoport>
</csoport>

<csoport csoport_id="3" produkcio="3">
 <tagok_szama>20</tagok_szama>
 <nev>Latin Ritmusok</nev>
 <egyesulet>Tánciskola</egyesulet>
 <korcsoport>25-30</korcsoport>
</csoport>

<csoport csoport_id="4" produkcio="4">
 <tagok_szama>12</tagok_szama>
 <nev>Jazz Express</nev>
 <egyesulet>Művészeti Társulat</egyesulet>
 <korcsoport>18-22</korcsoport>

</csoport>

<csoport csoport_id="5" produkció="5">

<tagok_szama>18</tagok_szama>

<nev>Flamenco Szenvedély</nev>

<egyesület>Táncműhely</egyesület>

<korcsoport>25-30</korcsoport>

</csoport>

<!--Versenyzők-->

<versenyző versenyző_id="1" tag="1">

<életkor>25</életkor>

<nev>Kovács Anna</nev>

<születési_datum>1998-04-10</születési_datum>

<lakcím>

<irányítószám>1024</irányítószám>

<város>Budapest</város>

<utca_háza szám>Kossuth utca 10</utca_háza szám>

</lakcím>

</versenyző>

<versenyző versenyző_id="2" tag="3">

<életkor>30</életkor>

<nev>Szabó Bence</nev>

<születési_datum>1992-08-22</születési_datum>

<lakcím>

<irányítószám>1132</irányítószám>

<város>Budapest</város>

<utca_háza szám>Rákospatak utca 5</utca_háza szám>

</lakcím>

</versenyző>

<versenyző versenyző_id="3" tag="4">

<életkor>22</életkor>

<nev>Molnár Rita</nev>

<születési_datum>2001-05-28</születési_datum>

<lakcím>

<irányítószám>3521</irányítószám>

<város>Debrecen</város>

<utca_háza szám>Sport utca 8</utca_háza szám>

</lakcím>

</versenyző>

<versenyző versenyző_id="4" tag="1">

<életkor>28</életkor>

<nev>Nagy Ferenc</nev>

<születési_datum>1995-11-15</születési_datum>

<lakcím>

```
        <iranyitoszam>4023</iranyitoszam>
        <varos>Debrecen</varos>
        <utca_hazszam>Sport tér 15</utca_hazszam>
    </lakcim>
</versenyzo>

<versenyzo versenyzo_id="5" tag="4">
    <eletkor>19</eletkor>
    <nev>Kiss Mária</nev>
    <szuletesi_datum>2004-07-03</szuletesi_datum>
    <lakcim>
        <iranyitoszam>1015</iranyitoszam>
        <varos>Budapest</varos>
        <utca_hazszam>Sport út 20</utca_hazszam>
    </lakcim>
</versenyzo>

<versenyzo versenyzo_id="6" tag="2">
    <eletkor>14</eletkor>
    <nev>Nagy Éva</nev>
    <szuletesi_datum>2009-09-05</szuletesi_datum>
    <lakcim>
        <iranyitoszam>4400</iranyitoszam>
        <varos>Nyíregyháza</varos>
        <utca_hazszam>Kossuth tér 5</utca_hazszam>
    </lakcim>
</versenyzo>

<versenyzo versenyzo_id="7" tag="3">
    <eletkor>26</eletkor>
    <nev>Kovács Péter</nev>
    <szuletesi_datum>1997-12-18</szuletesi_datum>
    <lakcim>
        <iranyitoszam>3525</iranyitoszam>
        <varos>Debrecen</varos>
        <utca_hazszam>Kisfaludy utca 12</utca_hazszam>
    </lakcim>
</versenyzo>

<versenyzo versenyzo_id="8" tag="3">
    <eletkor>29</eletkor>
    <nev>Szabó József</nev>
    <szuletesi_datum>1993-06-30</szuletesi_datum>
    <lakcim>
        <iranyitoszam>1139</iranyitoszam>
        <varos>Budapest</varos>
        <utca_hazszam>Kiss János utca 8</utca_hazszam>
    </lakcim>
</versenyzo>
```

```
<versenyzo versenyzo_id="9" tag="2">
  <nev>Szűcs László</nev>
  <szuletesi_datum>2010-02-14</szuletesi_datum>
  <lakcim>
    <iranyitoszam>4405</iranyitoszam>
    <varos>Nyíregyháza</varos>
    <utca_hazszam>Petőfi tér 3</utca_hazszam>
  </lakcim>
</versenyzo>
```

```
<versenyzo versenyzo_id="10" tag="2">
  <nev>Balogh Judit</nev>
  <szuletesi_datum>2008-11-10</szuletesi_datum>
  <lakcim>
    <iranyitoszam>3522</iranyitoszam>
    <varos>Debrecen</varos>
    <utca_hazszam>Bajnai út 7</utca_hazszam>
  </lakcim>
</versenyzo>
```

```
<!--A tánc helye több-több kapcsolat-->
```

```
<tanc_helye th_id="1" tanc_id="1" verseny_id="1">
  <pontszam>78</pontszam>
</tanc_helye>
```

```
<tanc_helye th_id="2" tanc_id="2" verseny_id="1">
  <pontszam>86</pontszam>
</tanc_helye>
```

```
<tanc_helye th_id="3" tanc_id="3" verseny_id="5">
  <pontszam>58</pontszam>
</tanc_helye>
```

```
<tanc_helye th_id="4" tanc_id="4" verseny_id="3">
  <pontszam>37</pontszam>
</tanc_helye>
```

```
<tanc_helye th_id="5" tanc_id="5" verseny_id="2">
  <pontszam>64</pontszam>
</tanc_helye>
```

```
<tanc_helye th_id="6" tanc_id="1" verseny_id="4">
  <pontszam>75</pontszam>
</tanc_helye>
```

```
<tanc_helye th_id="7" tanc_id="3" verseny_id="1">
  <pontszam>46</pontszam>
```

```

</tanc_helye>

<tanc_helye th_id="8" tanc_id="5" verseny_id="3">
  <pontszam>86</pontszam>
</tanc_helye>

<tanc_helye th_id="9" tanc_id="2" verseny_id="2">
  <pontszam>57</pontszam>
</tanc_helye>

<tanc_helye th_id="10" tanc_id="4" verseny_id="1">
  <pontszam>87</pontszam>
</tanc_helye>

<!--Lejátszott zene több-több kapcsolat-->

<lejatszott_zene tanc_id="1" zene_id="4"/>
<lejatszott_zene tanc_id="2" zene_id="3"/>
<lejatszott_zene tanc_id="4" zene_id="2"/>
<lejatszott_zene tanc_id="5" zene_id="5"/>
<lejatszott_zene tanc_id="3" zene_id="1"/>

</F4IIYA_Tancverseny>

```

4. XMLSchema készítése

Az XML dokumentumhoz való validáció segítése érdekében először egyszerű elemeket hoztam létre, amelyekre később hivatkozhatok a sémában. Ezután definiáltam ezekhez tartozó saját típusokat, köztük olyanokat, amelyeknél a típus korlátozása regex segítségével vagy enumerationnal történt. Ezt követően kidolgoztam magát a sémát, ahol az egyszerű típusokra történő hivatkozások mellett meghatároztam a minimális és maximális előfordulást is. A complexType-ok után részleteztem az attribútumokat, majd ezekhez létrehoztam a kulcsokat, idegenkulcsokat és speciális kapcsolat referenciákat.

Az XMLSchema kódja

```

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

  <!--Egyszerű elemek-->

  <xs:element name="cim" type="xs:string" />
  <xs:element name="hossz" type="hosszTipus" />
  <xs:element name="mufaj" type="mufajTipus" />
  <xs:element name="eloado" type="xs:string" />

  <xs:element name="nev" type="xs:string" />
  <xs:element name="terulet" type="xs:string" />
  <xs:element name="befogadokepesseg" type="xs:int" />
  <xs:element name="iranyitoszam" type="xs:int" />

```

```

<xs:element name="varos" type="xs:string" />
<xs:element name="utca_hazszam" type="xs:string" />

<xs:element name="t_mufaj" type="xs:string" />

<xs:element name="tagok_szama" type="xs:int" />
<xs:element name="egyesulet" type="xs:string" />
<xs:element name="korcsoport" type="xs:string" />

<xs:element name="eletkor" type="xs:int" />
<xs:element name="szuletesi_datum" type="idoTipus" />

<xs:element name="pontszam" type="xs:int" />

<!--Saját típusok-->

<xs:simpleType name="mufajTipus">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Pop" />
    <xs:enumeration value="Rock" />
    <xs:enumeration value="Elektronikus" />
    <xs:enumeration value="Romantikus" />
    <xs:enumeration value="Disco" />
    <xs:enumeration value="Jazz" />
    <xs:enumeration value="Country" />
    <xs:enumeration value="Hip-hop" />
    <xs:enumeration value="Rap" />
    <xs:enumeration value="Klasszikus" />
    <xs:enumeration value="Folk" />
    <xs:enumeration value="R and B " />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="hosszTipus">
  <xs:restriction base="xs:string">
    <xs:pattern value="[0-9]:[0-9][0-9]" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="idoTipus">
  <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>

<!--Felépítés-->

<xs:element name="F4IIYA_Tancverseny">

```

```

<xs:complexType>
  <xs:sequence>
    <xs:element name="zene" minOccurs="1" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="cim" />
          <xs:element ref="hossz" />
          <xs:element ref="mufaj" />
          <xs:element ref="eloado" />
        </xs:sequence>
        <xs:attribute name="zene_id" type="xs:int" />
      </xs:complexType>
    </xs:element>
    <xs:element name="versenyhelyszin" minOccurs="1"
maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="nev" />
          <xs:element ref="terulet" />
          <xs:element ref="befogadokepesseg" />
          <xs:element name="cim" minOccurs="1" maxOccurs="1">
            <xs:complexType>
              <xs:sequence>
                <xs:element ref="iranyitoszam" />
                <xs:element ref="varos" />
                <xs:element ref="utca_hazszam" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
        <xs:attribute name="verseny_id" type="xs:int" />
      </xs:complexType>
    </xs:element>
    <xs:element name="tanc" minOccurs="1" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="t_mufaj"/>
          <xs:element ref="nev" />
          <xs:element ref="hossz" />
        </xs:sequence>
        <xs:attribute name="tanc_id" type="xs:int" />
      </xs:complexType>
    </xs:element>
    <xs:element name="csoport" minOccurs="1"
maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="tagok_szama" />
          <xs:element ref="nev" />

```

```

        <xs:element ref="egyesulet" />
        <xs:element ref="korcsoport" minOccurs="1"
maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="csoport_id" type="xs:int" />
    <xs:attribute name="produkcio" type="xs:int" />
</xs:complexType>
</xs:element>
    <xs:element name="versenyzo" minOccurs="1"
maxOccurs="unbounded">
    <xs:complexType>
    <xs:sequence>
        <xs:element ref="eletkor" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element ref="nev" />
        <xs:element ref="szuletesi_datum" />
        <xs:element name="lakcim" minOccurs="1"
maxOccurs="1">
            <xs:complexType>
            <xs:sequence>
                <xs:element ref="iranyitoszam" />
                <xs:element ref="varos" />
                <xs:element ref="utca_hazszam" />
            </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
    <xs:attribute name="versenyzo_id" type="xs:int" />
    <xs:attribute name="tag" type="xs:int" />
    </xs:complexType>
</xs:element>
    <xs:element name="tanc_helye" minOccurs="1"
maxOccurs="unbounded">
    <xs:complexType>
    <xs:sequence>
        <xs:element ref="pontszam" />
    </xs:sequence>
    <xs:attribute name="th_id" type="xs:int" />
    <xs:attribute name="tanc_id" type="xs:int" />
    <xs:attribute name="verseny_id" type="xs:int" />
    </xs:complexType>
</xs:element>
    <xs:element name="lejatszott_zene" minOccurs="1"
maxOccurs="unbounded">
    <xs:complexType>
    <xs:attribute name="tanc_id" type="xs:int" />
    <xs:attribute name="zene_id" type="xs:int" />
    </xs:complexType>
</xs:element>

```

```

        </xs:sequence>
</xs:complexType>

<!--Kulcsok-->

<xs:key name="zene_kulcs">
    <xs:selector xpath="zene" />
    <xs:field xpath="@zene_id" />
</xs:key>

<xs:key name="helyszin_kulcs">
    <xs:selector xpath="versenyhelyszin" />
    <xs:field xpath="@verseny_id" />
</xs:key>

<xs:key name="tanc_kulcs">
    <xs:selector xpath="tanc" />
    <xs:field xpath="@tanc_id" />
</xs:key>

<xs:key name="csoport_kulcs">
    <xs:selector xpath="csoport" />
    <xs:field xpath="@csoport_id" />
</xs:key>

<xs:key name="versenyzo_kulcs">
    <xs:selector xpath="versenyzo" />
    <xs:field xpath="@versenyzo_id" />
</xs:key>

<xs:key name="tanc_helye_kulcs">
    <xs:selector xpath="tanc_helye" />
    <xs:field xpath="@th_id" />
</xs:key>

<!--Idegen kulcsok-->

<xs:keyref refer="tanc_kulcs" name="tanc_idegen_kulcs">
    <xs:selector xpath="csoport" />
    <xs:field xpath="@produkcio" />
</xs:keyref>

<xs:keyref refer="csoport_kulcs" name="csoport_idegen_kulcs">
    <xs:selector xpath="versenyzo" />
    <xs:field xpath="@tag" />
</xs:keyref>

<xs:keyref refer="tanc_kulcs" name="tanc_tanchelye_idegen_kulcs">
    <xs:selector xpath="tanc_helye" />

```



```

        <xs:field xpath="@tanc_id" />
    </xs:keyref>

    <xs:keyref                                refer="helyszin_kulcs"
name="verseny_tanchelye_idegen_kulcs">
        <xs:selector xpath="tanc_helye" />
        <xs:field xpath="@verseny_id" />
    </xs:keyref>

    <xs:keyref                                refer="tanc_kulcs"
name="tanc_lejatszottzene_idegen_kulcs">
        <xs:selector xpath="lejatszott_zene" />
        <xs:field xpath="@tanc_id" />
    </xs:keyref>

    <xs:keyref                                refer="zene_kulcs"
name="zene_lejatszottzene_idegen_kulcs">
        <xs:selector xpath="lejatszott_zene" />
        <xs:field xpath="@zene_id" />
    </xs:keyref>

    <!--1:1-->

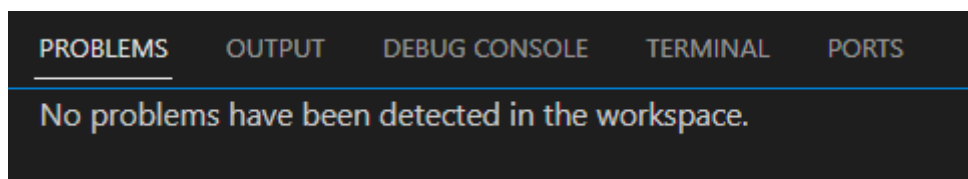
    <xs:unique name="unique_tanc">
        <xs:selector xpath="csoport" />
        <xs:field xpath="@produkcio" />
    </xs:unique>

</xs:element>

</xs:schema>

```

Validálás sikeressége:



5. DOM adatolvasás

Kezdetben a Java kódban megadom, hogy melyik fájlt kell beolvasni, és ezt a fájlt a projekt mappájában helyeztem el. Ezután létrehozok egy `Document` objektumot, amely segítségével lekérdezhetem a gyökérelemet, valamint a szülő- és gyerekelemeket is. A `NodeList` segítségével tárolni tudom a többszöri előfordulású elemeket, amiket a `document.getElementsByTagName(String)` segítségével tudok lekérdezni. Miután ezt megtettem, egy for ciklussal iterálok az adott szülőelemeken, hogy az attribútumokat és gyerekelemeket elmenthessem egy-egy stringbe, majd kiíráthassam azokat. Azoknál

az elemeknél, ahol több érték is lehet, ellenőrzöm, hogy az elemek száma több-e, mint a nem többszöri előfordulása eset. Ha igen, akkor ciklussal iterálok és kiíratom a több értéket.

A kód:

```
package DOMPARSEF4IIYA;

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 DOMReadF4IIYA {

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

        File xmlFile = new File("XML_F4IIYA.xml");

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

        Document doc = dBuilder.parse(xmlFile);

        doc.getDocumentElement().normalize();

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

        NodeList zList = doc.getElementsByTagName("zene");

        for (int i = 0; i < zList.getLength(); i++) {

            Node nNode = zList.item(i);
            System.out.println("\nCurrent element: " + nNode.getNodeName());

            if (nNode.getNodeType() == Node.ELEMENT_NODE) {

                Element elem = (Element) nNode;
                String id = elem.getAttribute("zene_id");
```

```

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

Node node2 = elem.getElementsByTagName("hossz").item(0);
String hossz = node2.getTextContent();

Node node3 = elem.getElementsByTagName("mufaj").item(0);
String mufaj = node3.getTextContent();

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

System.out.println("Zene id-je: " + id);
System.out.println("Cím: " + cim);
System.out.println("Hossz: " + hossz);
System.out.println("Műfaj: " + mufaj);
System.out.println("Előadó: " + eloado);

    }
}

NodeList vhList = doc.getElementsByTagName("versenyhelyszin");

for (int i = 0; i < vhList.getLength(); i++) {

    Node fNode = vhList.item(i);
    System.out.println("\nCurrent element: " + fNode.getNodeName());

    if (fNode.getNodeType() == Node.ELEMENT_NODE) {

        Element elem = (Element) fNode;
        String id = elem.getAttribute("verseny_id");

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

        Node node2 = elem.getElementsByTagName("terulet").item(0);
        String terulet = node2.getTextContent();

        Node node3 = elem.getElementsByTagName("befogadokepesseg").item(0);
        String befkep = node3.getTextContent();

        System.out.println("A helyszín id-je: " + id);
        System.out.println("A helyszín neve: " + nev);
        System.out.println("A helyszín területe: " + terulet);
        System.out.println("A helyszín befogadóképessége: " + befkep);

        if (vhList.item(i).getChildNodes().getLength() > 4) {

```

```

        int db = 0;
        Node node4 = elem.getElementsByTagName("cim").item(0);
        while (node4 != null) {
            node4 = elem.getElementsByTagName("cim").item(db);
            if (node4 != null) {
                Node n =
elem.getElementsByTagName("iranyitoszam").item(db);
                String isz = n.getTextContent();
                System.out.println("Telephely irányítószáma: " +
isz);

                Node n2 =
elem.getElementsByTagName("varos").item(db);
                String v = n2.getTextContent();
                System.out.println("Telephely városa: " + v);

                Node n3 =
elem.getElementsByTagName("utca_hazszam").item(db);
                String u = n3.getTextContent();
                System.out.println("Telephely utcája és házszáma:
" + u);
            }
            db++;
        }

    }
}

NodeList tList = doc.getElementsByTagName("tanc");

for (int i = 0; i < tList.getLength(); i++) {

    Node tNode = tList.item(i);
    System.out.println("\nCurrent element: " + tNode.getNodeName());

    if (tNode.getNodeType() == Node.ELEMENT_NODE) {

        Element elem = (Element) tNode;
        String tid = elem.getAttribute("tanc_id");

        Node node1 = elem.getElementsByTagName("t_mufaj").item(0);
        String mufaj = node1.getTextContent();

        Node node2 = elem.getElementsByTagName("nev").item(0);
        String nev = node2.getTextContent();
    }
}

```



```

        System.out.println("A korcsoport, amiben indul a
csoport: " + kcs);
    }
    db++;
}

}

}

}

NodeList vList = doc.getElementsByTagName("versenyzo");

for (int i = 0; i < vList.getLength(); i++) {

    Node gyNode = vList.item(i);
    System.out.println("\nCurrent element: " + gyNode.getNodeName());

    if (gyNode.getNodeType() == Node.ELEMENT_NODE) {

        Element elem = (Element) gyNode;
        String id = elem.getAttribute("versenyzo_id");
        String tag = elem.getAttribute("tag");

        Node node1 = null;
        String eletkor = null;
        if (elem.getElementsByTagName("eletkor").item(0) != null) {
            node1 = elem.getElementsByTagName("eletkor").item(0);
            eletkor = node1.getTextContent();
        }

        Node node2 = elem.getElementsByTagName("nev").item(0);
        String nev = node2.getTextContent();

        Node node3 =
elem.getElementsByTagName("szuletesi_datum").item(0);
        String szul = node3.getTextContent();

        System.out.println("Versenyző egyedi azonosítója: " + id);
        System.out.println("A csoport, ahol tag: " + tag);
        if (node1 != null)
            System.out.println("Életkor: " + eletkor);
        System.out.println("A versenyző neve: " + nev);
        System.out.println("A versenyző születési ideje: " + szul);

        if (vList.item(i).getChildNodes().getLength() > 4) {
            int db = 0;
            Node node4 = elem.getElementsByTagName("lakcim").item(0);
            while (node4 != null) {

```

```

        node4 = elem.getElementsByTagName("lakcim").item(db);
        if (node4 != null) {
            Node n =
elem.getElementsByTagName("iranyitoszam").item(db);
            String isz = n.getTextContent();
            System.out.println("lakcim irányítószáma: " +
isz);

            Node n2 =
elem.getElementsByTagName("varos").item(db);
            String v = n2.getTextContent();
            System.out.println("lakcim városa: " + v);

            Node n3 =
elem.getElementsByTagName("utca_hazszam").item(db);
            String u = n3.getTextContent();
            System.out.println("lakcim utcája és házszáma: "
+ u);
        }
        db++;
    }

}

}

}

NodeList thList = doc.getElementsByTagName("tanc_helye");

for (int i = 0; i < thList.getLength(); i++) {

    Node lNode = thList.item(i);
    System.out.println("\nCurrent element: " + lNode.getNodeName());

    if (lNode.getNodeType() == Node.ELEMENT_NODE) {

        Element elem = (Element) lNode;
        String th_id = elem.getAttribute("th_id");
        String tid = elem.getAttribute("tanc_id");
        String vid = elem.getAttribute("verseny_id");

        Node node1 = elem.getElementsByTagName("pontszam").item(0);
        String psz = node1.getTextContent();

        System.out.println("A tánc helye kapcsolat egyedi azonosítója:
" + th_id);

        System.out.println("A tánc azonosítója: " + tid);
        System.out.println("A versenyhely azonosítója: " + vid);
    }
}

```

```

        System.out.println("A kapott pontszám az adott helyszínen: "
+ psz);

    }

}

NodeList lzList = doc.getElementsByTagName("lejatszott_zene");

for (int i = 0; i < lzList.getLength(); i++) {

    Node tnNode = lzList.item(i);
    System.out.println("\nCurrent element: " + tnNode.getNodeName());

    if (tnNode.getNodeType() == Node.ELEMENT_NODE) {

        Element elem = (Element) tnNode;
        String tid = elem.getAttribute("tanc_id");
        String zid = elem.getAttribute("zene_id");

        System.out.println("A tánc azonosítója: " + tid);
        System.out.println("A zene azonosítója: " + zid);

    }

}

}

}

```

6. DOM adatmódosítás

Kiválasztottam, hogy az első csoport attribútumát szeretném módosítani, illetve egy gyerekelemét a nevet. Mivel ez az attribútum több helyen is megjelent, pl a versenyzőknél is, ezért ezeket ott is meg kellett változtatnom. A módosított fájlt mentettem, illetve kiírtattam a konzolra.

```

package DOMPARSEF4IIIYA;

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

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

```



```

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

public class DOMModifyF4IIYA {

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

        try {

            File inputFile = new File("XML_F4IIYA2.xml");

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

            Document doc = documentBuilder.parse(inputFile);

            // csoport attribútumának módosítása
            Node csoport = doc.getElementsByTagName("csoport").item(0);

            NamedNodeMap attr = csoport.getAttributes();
            Node nodeAttr = attr.getNamedItem("csoport_id");
            nodeAttr.setTextContent("6");

            // csoportnév módosítása
            NodeList list = csoport.getChildNodes();

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

                if (node.getNodeType() == Node.ELEMENT_NODE) {
                    Element eElement = (Element) node;

                    if ("nev".equals(eElement.getNodeName())) {
                        if ("Arany
Csillagok".equals(eElement.getTextContent())) {
                            eElement.setTextContent("Phoenix Fusion");
                        }
                    }
                }
            }
        }
    }
}

```

```

// Versenyzőknél való módosítás
NodeList vsList = doc.getElementsByTagName("versenyzo");

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

    if (node.getNodeType() == Node.ELEMENT_NODE) {

        Element eElement = (Element) node;
        String tagValue = eElement.getAttribute("tag");

        if ("1".equals(tagValue)) {
            eElement.setAttribute("tag", "6");
        }

    }
}

// Tartalom konzolra, illetve fájlba való írása
TransformerFactory transformerFactory =
TransformerFactory.newInstance();
Transformer transformer = transformerFactory.newTransformer();

DOMSource source = new DOMSource(doc);

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

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

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

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

```

7. DOM adatlekérdezés

A lekérdezéseket az XPath segítségével hajtottam végre, ebből 9-et csináltam, majd ezeket teszteltem, a kritériumoknak megfelelőket kiíratam a konzolra.

- a F4IYA_Tancverseny root elem zene gyerekelemeinek lekérdezése
- 2-es azonosítójú helyszín lekérdezése
- a második tánc kiválasztása

- csoportok, akik indulnak a 25-30as kategóriában
- csoportok, akik indulnak a 25-30as kategóriában, azoknak az egyesülete
- versenyzők születési dátumának és életkorának kiírása
- tanc_helye kapcsolat első két eleme
- versenyzők, akik több mint 23 évesek
- táncok, amik rendelkeznek bármilyen attribútummal

Kód:

```
package DOMPARSEF4IIYA;

import java.io.IOException;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.xpath.XPath;
import javax.xml.xpath.XPathConstants;
import javax.xml.xpath.XPathExpressionException;
import javax.xml.xpath.XPathFactory;

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 DOMQueryF4IIYA {

    public static void main(String[] args) {

        try {

            // DocumentBuilder létrehozása
            DocumentBuilderFactory documentBuilderFactory =
DocumentBuilderFactory.newInstance();
            DocumentBuilder documentBuilder =
documentBuilderFactory.newDocumentBuilder();

            Document document = documentBuilder.parse("XML_F4IIYA.xml");

            document.getDocumentElement().normalize();

            // az XPath készítése
            XPath xPath = XPathFactory.newInstance().newXPath();

            // meg kell adni az elérési út kifejezést és a csomópont listát

            // a F4IIYA_Tancverseny root elem zene gyerekelemeinek lekérdezése
```

```

String expression = "F4IIYA_Tancverseny / zene";

// 2-es azonosítóju helyszin lekerdezese
// String expression = "//versenyhelyszin[@verseny_id='2']";

// a masodik tanc kivalasztasa
// String expression = "F4IIYA_Tancverseny/tanc[2]";

// csoportok, akik indulnak a 25-30as kategoriaban
// String expression = "//csoport[korcsoport='25-30']";

//csoportok, akik indulnak a 25-30as kategoriaban, azoknak az
egyesulete
// String expression = "//csoport[korcsoport='25-30']/egyesulet";

// versenyzok szuletesi datumanak es életkorának kiiratasa
// String expression = "//szuletesi_datum | //etekor";

// tanc_helye kapcsolat elso ket eleme
// String expression = "//tanc_helye[position()<3]";

// versenyzok, akik tobb mint 23 evesek
// String expression = "//versenyzo[etekor>23]/nev";

// tancok, amik rendelkeznek barmilyen attributummal
// String expression = "//tanc[@*]";

// keszitunk egy listat, majd az XPath kifejezest le kell fordítani
es ki kell ertekelni
NodeList nodeList = (NodeList)
XPath.compile(expression).evaluate(document, XPathConstants.NODESET);

// A for ciklus segitsegevel a NodeList csomopontjain vegig kell
iterálni
for (int i = 0; i < nodeList.getLength(); i++) {
    Node node = nodeList.item(i);

    System.out.println("\nAktualis elem: " + node.getNodeName());

    // Meg kell vizsgalni a csomopontot, tesztelni kell a
subelementet, jelen esetben a zene elemet
    if (node.getNodeType() == Node.ELEMENT_NODE &&
node.getNodeName().equals("zene")) {

        Element elem = (Element) node;

        String id = elem.getAttribute("zene_id");

        Node node1 = elem.getElementsByTagName("cim").item(0);

```

```

        String cim = node1.getTextContent();

        Node node2 =
elem.getElementsByTagName("hossz").item(0);
        String hossz = node2.getTextContent();

        Node node3 =
elem.getElementsByTagName("mufaj").item(0);
        String mufaj = node3.getTextContent();

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

        System.out.println("Zene id-je: " + id);
        System.out.println("Cím: " + cim);
        System.out.println("Hossz: " + hossz);
        System.out.println("Műfaj: " + mufaj);
        System.out.println("Előadó: " + eloado);

    }

    // Csoport egyesületének kiirratasa
    if (node.getNodeType() == Node.ELEMENT_NODE &&
node.getNodeName().equals("egyesulet")) {

        Element element = (Element) node;

        System.out.println("Egyesulet: " +
element.getTextContent());

    }

    // Csoport kiirratasa
    if (node.getNodeType() == Node.ELEMENT_NODE &&
node.getNodeName().equals("csoport")) {

        Element element = (Element) node;

        System.out.println("ID: " +
element.getAttribute("csoport_id"));

        System.out.println(
            "Helyszin neve: " +
element.getElementsByTagName("nev").item(0).getTextContent());

        System.out.println(
            "Terulete: " +
element.getElementsByTagName("tagok_szama").item(0).getTextContent());
    }

```

```

        System.out.println(
                                "Befogadokepesseg: " +
element.getElementsByTagName("egyesulet").item(0).getTextContent());

        System.out.println(
                                "Korcsoport: " +
element.getElementsByTagName("korcsoport").item(0).getTextContent());

    }

    // Versenyző születési idejének kiíratása
    if (node.getNodeType() == Node.ELEMENT_NODE &&
node.getNodeName().equals("szuletési_datum")) {

        Element element = (Element) node;

        System.out.println("Születési datum: " +
element.getTextContent());

    }

    // Versenyző életkorának kiíratása
    if (node.getNodeType() == Node.ELEMENT_NODE &&
node.getNodeName().equals("eletkor")) {

        Element element = (Element) node;

        System.out.println("Életkor: " +
element.getTextContent());

    }

    // Versenyző nevének kiíratása
    if (node.getNodeType() == Node.ELEMENT_NODE &&
node.getNodeName().equals("nev")) {

        Element element = (Element) node;

        System.out.println("Név: " + element.getTextContent());

    }

    //Versenyhelyszín kiíratása
    if (node.getNodeType() == Node.ELEMENT_NODE &&
node.getNodeName().equals("versenyhelyszin")) {

        Element element = (Element) node;

```

```

                                System.out.println("ID: " +
element.getAttribute("verseny_id"));

        System.out.println(
                                "Helyszin neve: " +
element.getElementsByTagName("nev").item(0).getTextContent());

        System.out.println(
                                "Terulete: " +
element.getElementsByTagName("terulet").item(0).getTextContent());

        System.out.println(
                                "Befogadokepesseg: " +
element.getElementsByTagName("befogadokepesseg").item(0).getTextContent());

        if (nodeList.item(i).getChildNodes().getLength() > 4) {
            int db = 0;

                                Node node3 =
element.getElementsByTagName("cim").item(0);
            while (node3 != null) {
                                node3 =
element.getElementsByTagName("cim").item(db);
                if (node3 != null) {
                                Node n =
element.getElementsByTagName("iranyitoszam").item(db);
                    String isz = n.getTextContent();
                    System.out.println("Telephely iranyitoszama:
" + isz);
                                Node n2 =
element.getElementsByTagName("varos").item(db);
                    String v = n2.getTextContent();
                    System.out.println("Telephely varosa: " +
v);
                                Node n3 =
element.getElementsByTagName("utca_hazszam").item(db);
                    String u = n3.getTextContent();
                    System.out.println("Telephely utca,hazszama:
" + u);
                }
                db++;
            }
        }
    }

    //tanc kiiratasa
    if (node.getNodeType() == Node.ELEMENT_NODE &&
node.getNodeName().equals("tanc")) {

```

```

        Element element = (Element) node;

        System.out.println("ID: " +
element.getAttribute("tanc_id"));

        System.out.println("Nev: " +
element.getElementsByTagName("nev").item(0).getTextContent());

        System.out.println(
                                "Mufaj: " +
element.getElementsByTagName("t_mufaj").item(0).getTextContent());

        System.out.println(
                                "hossz: " +
element.getElementsByTagName("hossz").item(0).getTextContent());

    }

    //tanc_hely kiiratas
    if (node.getNodeType() == Node.ELEMENT_NODE &&
node.getNodeName().equals("tanc_helye")) {

        Element element = (Element) node;

        System.out.println("Tanc id: " +
element.getAttribute("tanc_id"));

        System.out.println("Elem id: " +
element.getAttribute("th_id"));

        System.out.println("Versenyhelyszin id: " +
element.getAttribute("verseny_id"));

        System.out
                                .println("Pontszam: " +
element.getElementsByTagName("pontszam").item(0).getTextContent());

    }

}

} catch (ParserConfigurationException e) {
    e.printStackTrace();
} catch (SAXException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}

```



```

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

    }

}

```

8. DOM adatírás

Minden egyes root gyerekelemnek létrehoztam egy-egy metódust aminek segítségével hozzá tudom adni ezeket a root-hoz. A metódusok paraméterei az attribútumok, illetve az egyes elemek tulajdonságai, amiket ezen belül hozzáadok az elemhez. Mindegyik elemhez létrehozok legalább hármat, amely megjelenik az xml dokumentumban is. Miután ezt létrehoztam, megnéztem, hogy az XSD-vel való validálás során minden rendben van –e, és ez így volt.

Kód:

```

package DOMPARSEF4IIYA;

import java.io.File;

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 javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;

import org.w3c.dom.Comment;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;

public class DOMWriteF4IIYA {

    public static void main(String argv[]) throws Exception {

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

        Document doc = dBuilder.newDocument();

        Element root = doc.createElementNS("XMLF4IIYA",
"F4IIYA_Tancverseny");
    }
}

```

```

doc.appendChild(root);

// zenek

root.appendChild(createZene(doc, "1", "Cim 1", "3:45", "Pop", "Kovács
Anna"));
        root.appendChild(createZene(doc, "2", "Cim 2", "2:45",
"Elektronikus", "Tiesto"));
        root.appendChild(createZene(doc, "3", "Cim 3", "3:29", "Folk", "Szabó
István"));

Element element = (Element) doc.getElementsByTagName("zene").item(0);
Comment comment = doc.createComment("Zenék");
element.getParentNode().insertBefore(comment, element);

// versenyhelyszin

root.appendChild(createVersenyhelyszin(doc, "1", "Park Stadion", "800
nm^2", "5000", "1024", "Budapest", "Kossuth utca 10"));
        root.appendChild(createVersenyhelyszin(doc, "2", "Sportaréna", "1200
nm^2", "10000", "1132", "Budapest", "Rákospatak utca 5"));
        root.appendChild(createVersenyhelyszin(doc, "3", "Sportközpont", "600
nm^2", "3000", "3521", "Debrecen", "Sport utca 8"));

                                element                =                (Element)
doc.getElementsByTagName("versenyhelyszin").item(0);
        comment = doc.createComment("Versenyhelyszínek");
        element.getParentNode().insertBefore(comment, element);

// tancok

root.appendChild(createTanc(doc, "1", "Modern tánc", "Fények játéka",
"4:30"));
        root.appendChild(createTanc(doc, "2", "Latin tánc", "Forró ritmusok",
"3:15"));
        root.appendChild(createTanc(doc, "3", "Balett", "Hamupipőke álom",
"5:00"));

element = (Element) doc.getElementsByTagName("tanc").item(0);
comment = doc.createComment("Táncok");
element.getParentNode().insertBefore(comment, element);

// Csoportok

String[] kcs = {"25-30"};
        root.appendChild(createCsoport(doc, "1", "1", "15", "Arany
Csillagok", "Táncművészeti Egyesület", kcs));

String[] kcs2 = {"25-30", "22-25"};

```

```

        root.appendChild(createCsoport(doc, "2", "2", "10", "Latin Ritmusok",
"Tánciskola", kcs2));

        String[] kcs3 = {"12-14"};
        root.appendChild(createCsoport(doc, "3", "3", "14", "Kis Angyalok",
"Balett Stúdió", kcs3));

        element = (Element) doc.getElementsByTagName("csoport").item(0);
        comment = doc.createComment("Csoportok");
        element.getParentNode().insertBefore(comment, element);

        // Versenyzők

        root.appendChild(createVersenyzo(doc, "3", "1", "Kovács Anna", "2009-
04-10", "1024", "Budapest", "Kossuth utca 10", "14"));
        root.appendChild(createVersenyzo(doc, "2", "2", "Szabó Bence", "1992-
08-22", "1132", "Budapest", "Rákospatak utca 5", ""));
        root.appendChild(createVersenyzo(doc, "1", "3", "Nagy Ferenc", "1998-
04-10", "4023", "Debrecen", "Sport tér 15", "25"));

        element = (Element) doc.getElementsByTagName("versenyzo").item(0);
        comment = doc.createComment("Versenyzők");
        element.getParentNode().insertBefore(comment, element);

        // tanc_helye

        root.appendChild(createTancHelye(doc, "2", "1", "3", "78"));
        root.appendChild(createTancHelye(doc, "1", "2", "1", "82"));
        root.appendChild(createTancHelye(doc, "3", "3", "2", "76"));

        element = (Element) doc.getElementsByTagName("tanc_helye").item(0);
        comment = doc.createComment("A tánc helye több-több kapcsolat");
        element.getParentNode().insertBefore(comment, element);

        // lejjatszott_zene

        root.appendChild(createLejjatszottZene(doc, "1", "1"));
        root.appendChild(createLejjatszottZene(doc, "2", "2"));
        root.appendChild(createLejjatszottZene(doc, "3", "3"));

        element = (Element)
doc.getElementsByTagName("lejjatszott_zene").item(0);
        comment = doc.createComment("Lejjatszott zene több-több kapcsolat");
        element.getParentNode().insertBefore(comment, element);

        // Transform

        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");

// File letrehozás

DOMSource source = new DOMSource(doc);
File myFile = new File("XMLF4IIYA_write.xml");

// Konzolra való kiírás

//StreamResult console = new StreamResult(System.out);
StreamResult file = new StreamResult(myFile);

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

}

```

```

private static Node createZene(Document doc, String zene_id, String cim,
String hossz, String mufaj,
    String eloado) {

    Element z = doc.createElement("zene");

    z.setAttribute("zene_id", zene_id);
    z.appendChild(createElement(doc, "cim", cim));
    z.appendChild(createElement(doc, "hossz", hossz));
    z.appendChild(createElement(doc, "mufaj", mufaj));
    z.appendChild(createElement(doc, "eloado", eloado));

    return z;

}

```

```

private static Node createVersenyhelyszin(Document doc, String
verseny_id, String nev, String terület,
    String befogadokepesseg, String isz, String varos, String
utcahaszszam) {

    Element vh = doc.createElement("versenyhelyszin");

    vh.setAttribute("verseny_id", verseny_id);
    vh.appendChild(createElement(doc, "nev", nev));
    vh.appendChild(createElement(doc, "terulet", terület));

```

```
        vh.appendChild(createElement(doc, "befogadokepesseg",  
befogadokepesseg));
```

```
    Element cim = doc.createElement("cim");  
    cim.appendChild(createElement(doc, "iranyitoszam", isz));  
    cim.appendChild(createElement(doc, "varos", varos));  
    cim.appendChild(createElement(doc, "utca_hazszam", utcahazszam));
```

```
    vh.appendChild(cim);
```

```
    return vh;
```

```
}
```

```
    private static Node createTanc(Document doc, String tanc_id, String  
t_mufaj, String nev,  
        String hossz) {
```

```
    Element tr = doc.createElement("tanc");
```

```
    tr.setAttribute("tanc_id", tanc_id);  
    tr.appendChild(createElement(doc, "t_mufaj", t_mufaj));  
    tr.appendChild(createElement(doc, "nev", nev));  
    tr.appendChild(createElement(doc, "hossz", hossz));
```

```
    return tr;
```

```
}
```

```
    private static Node createCsoport(Document doc, String csoport_id, String  
produkcio, String tagok_szama,  
        String nev, String egyesulet, String[] korcsoport) {
```

```
    Element csop = doc.createElement("csoport");
```

```
    csop.setAttribute("csoport_id", csoport_id);  
    csop.setAttribute("produkcio", produkcio);  
    csop.appendChild(createElement(doc, "tagok_szama", tagok_szama));  
    csop.appendChild(createElement(doc, "nev", nev));  
    csop.appendChild(createElement(doc, "egyesulet", egyesulet));
```

```
    Node[] node1 = appendArray(doc, "korcsoport", korcsoport);
```

```
    for (int i = 0; i < korcsoport.length; i++) {  
        csop.appendChild(node1[i]);  
    }
```

```
    return csop;
```

```

    }

    private static Node createVersenyzo(Document doc, String tag, String
versenyzo_id, String nev, String szuletesi_datum,
        String isz, String varos, String utcahazszam, String eletkor) {

        Element v = doc.createElement("versenyzo");

        v.setAttribute("tag", tag);
        v.setAttribute("versenyzo_id", versenyzo_id);
        if(!(eletkor.equals(""))){
            v.appendChild(createElement(doc, "eletkor", eletkor));
        }
        v.appendChild(createElement(doc, "nev", nev));
        v.appendChild(createElement(doc, "szuletesi_datum",
szuletesi_datum));

        Element cim = doc.createElement("lakcim");
        cim.appendChild(createElement(doc, "iranyitoszam", isz));
        cim.appendChild(createElement(doc, "varos", varos));
        cim.appendChild(createElement(doc, "utca_hazszam", utcahazszam));

        v.appendChild(cim);

        return v;
    }

    private static Node createTancHelye(Document doc, String tanc_id, String
th_id, String verseny_id,
        String pontszam) {

        Element th = doc.createElement("tanc_helye");

        th.setAttribute("tanc_id", tanc_id);
        th.setAttribute("th_id", th_id);
        th.setAttribute("verseny_id", verseny_id);
        th.appendChild(createElement(doc, "pontszam", pontszam));

        return th;
    }

    private static Node createLejatszottZene(Document doc, String tanc_id,
String zene_id) {

        Element lz = doc.createElement("lejatszott_zene");

        lz.setAttribute("tanc_id", tanc_id);

```

```

        lz.setAttribute("zene_id", zene_id);

        return lz;
    }

    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[] appendArray(Document doc, String name, String[]
value) {

        Element nodes[] = new Element[value.length];

        for (int i = 0; i < value.length; i++) {

            nodes[i] = doc.createElement(name);
            nodes[i].appendChild(doc.createTextNode(value[i]));

        }

        return nodes;
    }

}

```