

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Kutyaiskola

Készítette: **Palencsár Enikő**

Neptunkód: **YD11NL**

Dátum: **2023. 12. 02.**

Tartalomjegyzék

Tartalomjegyzék	2
Bevezetés	3
A feladat leírása	3
1. feladat	4
Az adatbázis ER modell tervezése	4
Az adatbázis konvertálása XDM modellre	6
Az XDM modell alapján XML dokumentum készítése	7
Az XML dokumentum alapján XMLSchema készítése - saját típusok, ref, key, keyref, speciális elemek	11
2. feladat	16
Adatolvasás	16
Adatmódosítás	28
Adatlekérdezés	38
Adatírás	45

Bevezetés

A feladat leírása

A kutyaiskola című feladat keretein belül egy kisebb kertvárosi kutyaiskola adatainak nyilvántartását végzem el XML dokumentumban. A kutyaiskolába **kutyák** járnak, akiknek van nevük, fajtájuk, nemük és születési dátumuk. Ha ez utóbbi nem ismert, megkérjük az állatorvost, hogy adjon egy tippet a kutya korára, születési dátumára vonatkozóan, és ezt jegyezzük fel. A kutya neme kan vagy szuka lehet. Minden kutyához egyértelműen tartozik egy **tagság** adatlap, melyben eltároljuk, hogy mikor írárták be az adott kutyát a kutyaiskolába, hogy aktív-e jelenleg a tagsága, és hogy normál vagy kedvezményes tarifa vonatkozik rá. Kedvezmény jár például, ha egy gazdi több kutyáját is beírattja, vagy ha a gazdi állatmenhelyen önkénteskedik. Tárolásra kerülnek továbbá a **gazdik** adatai: a gazdi neve, születési éve – hiszen a szervezés szempontjából nem mindegy, hogy idős néni vagy fiatal úriember kíséri a kutyát a foglalkozásokra –, továbbá a gazdi elérhetőségei. Az elérhetőségek között szerepel legalább egy (legfeljebb 5) e-mail cím, melyre az aktuális foglalkozásokról szóló értesítést küldik, továbbá legalább egy telefonszám olyan esetekre, amikor a kutyaiskolának sürgősen el kell érnie a gazdit.

A kutyaiskola alkalmazottai a kutyakiképzők, azaz **trénerek**, ők tartják a foglalkozásokat. A trénereknek tároljuk a nevüket és az elérhetőségeiket a gazdikhoz hasonló szabályok szerint, valamint az órabérüket. Opcionálisan szakterületet is rendelhetünk hozzájuk (pl: activity, agility, kutyapszichológia), de ez nem kötelező. Ezenkívül nyilvántartjuk a kutyaiskola által meghirdetett **foglalkozásokat**, melyeknek van vezetője (egy tréner) és vannak résztvevői (kutyák). Minden foglalkozáshoz rendelünk egy vagy több általános témát, hiszen nem mindegy, hogy a gazdi engedelmességre oktatni, vagy őrző-védő foglalkozásra hozza be a kutyáját. A foglalkozásokhoz tartozik továbbá egy helyszín, azaz egy pálya, ahol a foglalkozást tartják, valamint egy időpont, melyet dátum, kezdet és vég tagokra bonthatunk. A kezdet és a vég egész órák lehetnek. Minden foglalkozás reggel 6 óra után kezdődik és este 21 óra előtt véget ér. Az egyes foglalkozásokon a jól viselkedő kutyák a kutyatrénertől kisebb jutalmat kaphatnak, ez lehet jutalomfalat, de akár játék is (kislabda, rágócsont stb.). A kiadott jutalmakat szintén eltároljuk az XML dokumentumban. A trénerek minden foglalkozás végén egy ötös skálán értékelik a gazdik kutyáik feletti kontrollját, ezt az értékelést szintén tartalmazni fogja az XML fájl.

1. feladat

1a)

Az adatbázis ER modell tervezése

A kutyaiskola ER modelljében öt egyed található, ezek az alábbiak: tréner, gazdi, foglalkozás, kutya és tagság. Minden egyednek van kulcsa, melyet aláhúzás jelöl. A kulcsok elnevezései az *egyednév kezdőbetű + kód* szabály szerint jöttek létre, kivéve a tagság esetén, ahol tagkód lett az egyedi azonosító neve, hiszen a tkódot a trénernél már elhasználtuk.

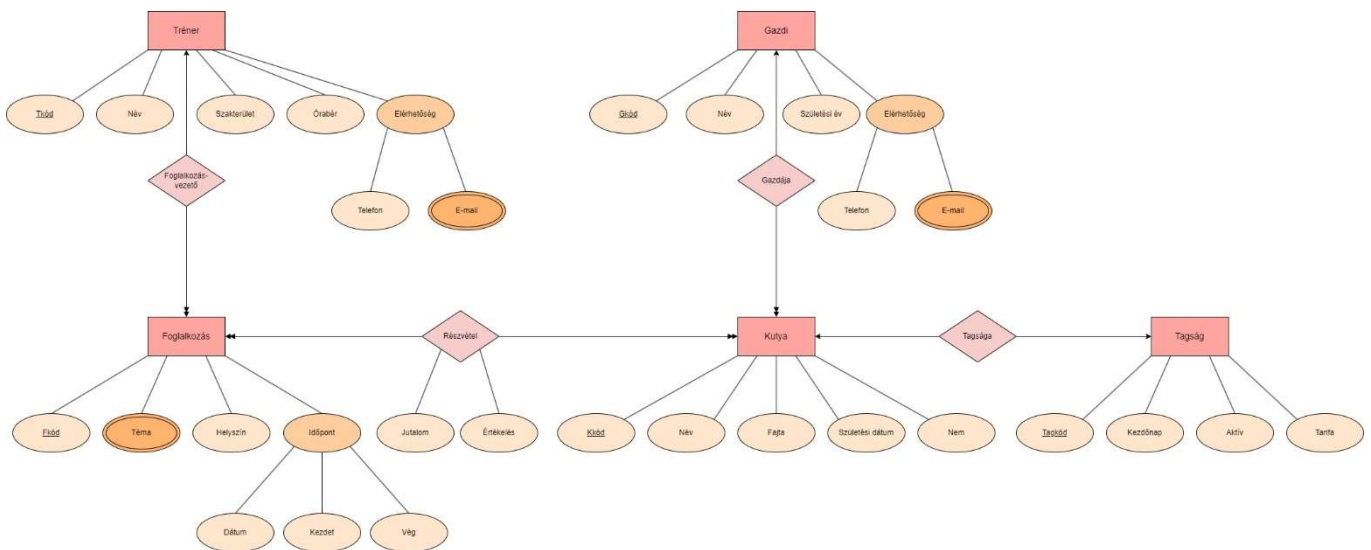
Kutya és tagság között 1:1, kutya és foglalkozás között N:M, míg gazdi és kutya, valamint tréner és foglalkozás között 1:N kapcsolat van. Ez azt jelenti, hogy egy kutyának pontosan egy tagsága van, és egy tagsághoz pontosan egy kutya tartozik; továbbá egy foglalkozáson több kutya is részt vehet, és egy kutya több foglalkozáson is megjelenhet. Egy kutyának egy gazdája van, de egy gazdinak lehet több kutyája is. Egy foglalkozást egy tréner vezet, de ugyanaz a tréner vezethet több foglalkozást is.

A modellben több összetett tulajdonságot láthatunk, ilyen a gazdi és a tréner esetén az elérhetőség, valamint a foglalkozások időpontja. Vannak többértékű tulajdonságok is, nevezetesen az e-mail, melyből egy embernek (gazdinak, trénernek) akár több is lehet, valamint a foglalkozások témái. A foglalkozás-kutya N:M kapcsolat tulajdonságai a jutalom, mely megadja, hogy adott foglalkozáson adott kutya milyen jutalmat kapott, továbbá az értékelés, ami a foglalkozást tartó tréner által adott pontszám a gazdi rátermettségére.

Egyedek és tulajdonságaik:

- Tréner
 - Tkód: egyedi azonosító, numerikus
 - Név: a tréner teljes neve
 - Szakterület: a tréner szakterülete, ha van
 - Órabér: a trénernek fizetett szerződéses órabér
 - Elérhetőség: a tréner telefonszáma és e-mail címei
- Gazdi
 - Gkód: egyedi azonosító, numerikus
 - Név: a gazdi teljes neve
 - Születési év: a gazdi születési éve
 - Elérhetőség: a gazdi telefonszáma és e-mail címei
- Foglalkozás
 - Fkód: egyedi azonosító, numerikus
 - Téma: a foglalkozás témájának rövid szöveges leírása
 - Helyszín: a foglalkozás helyszíne (például: gyakorlóaréna, akadálypálya)
 - Időpont: a foglalkozás dátuma, kezdete és vége (6:00 és 21:00 közötti egész órák)
- Kutya
 - Kkód: egyedi azonosító, numerikus
 - Név: a kutya neve
 - Fajta: a kutya fajtája
 - Születési dátum: a kutya születési dátuma, ha ismeretlen, akkor erre vonatkozó becslés
 - Nem: a kutya neme, kan vagy szuka lehet
 - Időpont: a foglalkozás dátuma, kezdete és vége (6:00 és 21:00 közötti egész órák)

- **Tagság**
 - Tagkód: egyedi azonosító, numerikus
 - Kezdőnap: a beiratkozás dátuma
 - Aktív: aktív-e a kutya tagsága, igen vagy nem lehet
 - Tarifa: normál vagy kedvezményes lehet, a foglalkozások díjára vonatkozik
- **Részvétel kapcsolat:**
 - Jutalom: a foglalkozáson kapott jutalom rövid szöveges leírása
 - Értékelés: a gazdi rátermettségének értékelése 1-5-ig



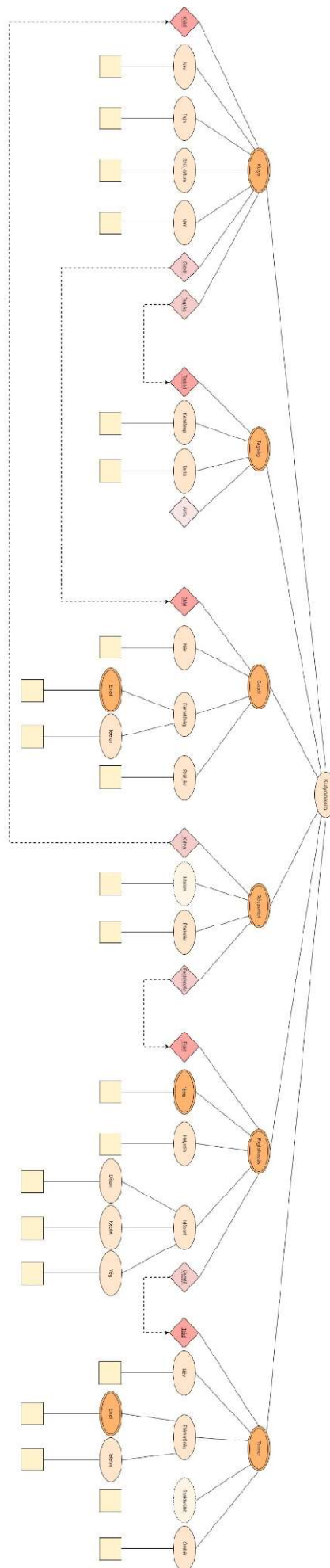
1b)

Az adatbázis konvertálása XDM modellre

Az adatbázis XDM modellje az ER modell alapján került megvalósításra. A gyökérelem neve kutyaaiskola lett, melynek 6 db többpéldányos gyerekeleme van. Minden gyerekelemnek van kulcs attribútuma, melyeket az ER modellben szereplő kulcs nevekkkel láttam el.

A kutya-tagság 1:1 kapcsolat jelölésére a kutya elemnek lett egy tagság nevű idegenkulcs attribútuma, ami a tagság elem elsődleges kulcsára (tagkód) mutat. Mivel a tagság aktív tulajdonsága igen primitív, elemi értékű, és csak igen-nem értékeket vehet fel, belőle is attribútum lett. A kutya-foglalkozás N:M kapcsolatot egy részvétel elnevezésű új elem jelöli, melyben található kutya és foglalkozás idegenkulcs is. A kutya-gazdi 1:N kapcsolat ábrázolására a kutya elemnek lett egy gazdi idegenkulcs attribútuma, ami a gazdi elem gkód kulcsértékére mutat. A tréner-foglalkozás 1:N kapcsolat reprezentálása is hasonlóan történt, ebben az esetben a foglalkozáshoz került a vezető nevű, tréner tkód értékére mutató idegenkulcs.

Többpéldányos elemek lettek azok az elemek is, melyek az ER modellben többértékű tulajdonságok voltak: az e-mail cím és a téma. A trénerek szakterülete, valamint a részvételben szereplő jutalom opcionális elemek, nem kötelezően szerepelnek, így az ezeket jelölő ellipszisek szegélye szaggatott. A hierarchiában legalul elhelyezkedő gyerekelemeknek szövegtartalma van, ezt téglalapok jelölik.



1c)

Az XDM modell alapján XML dokumentum készítése

Az XML dokumentumot az XDM modell alapján készítettem el. A gyökérelem így a kutyaiskola lett, a főbb egyedek pedig ennek többpéldányos gyermekelemei, melyekből típusonként legalább 3 különbözőt létrehoztam. A tagságok esetén az *aktív* attribútumot két helyen expliciten beállítottam, de ennek egyébként alapértelmezett értéke is van, ami az „igen”. Felvettem olyan gazdikat és trénereket, akiknek egynél több e-mail címük van, továbbá olyan foglalkozásokat, amiken több téma is érintésre került. A trénerek közül két személyhez adtam meg szakterületet, a harmadikhoz nem, hiszen ez egy opcionális gyerekelem. A részvétel adatokhoz sem adtam mindenhol jutalmat, mert az sem kötelező elem.

Az XML dokumentum tartalma:

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

<kutyaiskola xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
  xs:noNamespaceSchemaLocation="XMLSchemaYd11n1.xsd">

  <!--Kutyák, akik be vannak regisztrálva-->
  <kutya kkód="1" gazdi="11" tagság="1000">
    <név>Bizsu</név>
    <fajta>yorkshire terrier</fajta>
    <szül_dátum>2022-10-10</szül_dátum>
    <nem>kan</nem>
  </kutya>
  <kutya kkód="2" gazdi="11" tagság="1002">
    <név>Buksi</név>
    <fajta>németjuhász</fajta>
    <szül_dátum>2021-05-01</szül_dátum>
    <nem>kan</nem>
  </kutya>
  <kutya kkód="3" gazdi="12" tagság="1001">
    <név>Bella</név>
    <fajta>magyar vizsla</fajta>
    <szül_dátum>2023-04-23</szül_dátum>
    <nem>szuka</nem>
  </kutya>
  <kutya kkód="4" gazdi="13" tagság="1003">
    <név>Mister</név>
    <fajta>afgán agár</fajta>
    <szül_dátum>2016-09-15</szül_dátum>
    <nem>kan</nem>
  </kutya>

  <!--Kutyák tagság adatai-->
  <tagság tagkód="1000" aktív="igen">
    <kezdőnap>2023-03-01</kezdőnap>
    <tarifa>kedvezményes</tarifa>
  </tagság>
```

```

<tagság tagkód="1001">
  <kezdőnap>2023-10-01</kezdőnap>
  <tarifa>normál</tarifa>
</tagság>
<tagság tagkód="1002">
  <kezdőnap>2022-02-28</kezdőnap>
  <tarifa>normál</tarifa>
</tagság>
<tagság tagkód="1003" aktív="nem">
  <kezdőnap>2022-01-01</kezdőnap>
  <tarifa>kedvezményes</tarifa>
</tagság>

<!--Gazdik adatai-->
<gazdi gkód="11">
  <név>Kiss Katalin</név>
  <elérhetőség>
    <email>kisskat@gmail.com</email>
    <email>kati@kkconsulting.hu</email>
    <email>zoltan.kiss@gmail.com</email>
    <telefon>+36701234987</telefon>
  </elérhetőség>
  <szül_év>1980</szül_év>
</gazdi>
<gazdi gkód="12">
  <név>Kovács Béláné Folt Irén</név>
  <elérhetőség>
    <email>incike@citromail.hu</email>
    <telefon>+36208877665</telefon>
  </elérhetőség>
  <szül_év>1961</szül_év>
</gazdi>
<gazdi gkód="13">
  <név>Petőfi Mór</név>
  <elérhetőség>
    <email>borok.mortol@gmail.com</email>
    <email>talprabaradlay@gmail.com</email>
    <telefon>+36506060768</telefon>
  </elérhetőség>
  <szül_év>1972</szül_év>
</gazdi>

<!--Foglalkozások-->
<foglalkozás fkód="100" vezető="51">
  <téma>elsősegély oktatás</téma>
  <téma>kutyapszichológia előadás</téma>
  <téma>engedelmesség tréning</téma>
  <helyszín>gyakorlóaréna</helyszín>
  <időpont>
    <dátum>2023-10-01</dátum>

```



```

        <kezdet>14</kezdet>
        <vég>18</vég>
    </időpont>
</foglalkozás>
<foglalkozás fkód="101" vezető="53">
    <téma>agility foglalkozás</téma>
    <téma>activity foglalkozás</téma>
    <helyszín>akadálypálya</helyszín>
    <időpont>
        <dátum>2023-10-08</dátum>
        <kezdet>15</kezdet>
        <vég>16</vég>
    </időpont>
</foglalkozás>
<foglalkozás fkód="102" vezető="51">
    <téma>engedelmesség tréning</téma>
    <helyszín>akadálypálya</helyszín>
    <időpont>
        <dátum>2023-10-15</dátum>
        <kezdet>15</kezdet>
        <vég>17</vég>
    </időpont>
</foglalkozás>
<foglalkozás fkód="103" vezető="51">
    <téma>engedelmesség tréning</téma>
    <helyszín>akadálypálya</helyszín>
    <időpont>
        <dátum>2023-11-02</dátum>
        <kezdet>10</kezdet>
        <vég>12</vég>
    </időpont>
</foglalkozás>

<!--Kutya-foglalkozás kapcsoló elemek avagy részvételi adatok-->
<részvétel kutya="1" foglalkozás="100">
    <értékelés>4</értékelés>
</részvétel>
<részvétel kutya="2" foglalkozás="100">
    <értékelés>5</értékelés>
    <jutalom>Rágócsont</jutalom>
</részvétel>
<részvétel kutya="2" foglalkozás="101">
    <értékelés>5</értékelés>
    <jutalom>Purina száraz kutyatáp</jutalom>
</részvétel>
<részvétel kutya="3" foglalkozás="101">
    <értékelés>2</értékelés>
    <jutalom>Rágócsont</jutalom>
</részvétel>
<részvétel kutya="1" foglalkozás="102">

```

```

        <értékelés>5</értékelés>
        <jutalom>Kislabda</jutalom>
    </részvétel>
    <részvétel kutya="2" foglalkozás="102">
        <értékelés>5</értékelés>
        <jutalom>Játzókötel</jutalom>
    </részvétel>
    <részvétel kutya="1" foglalkozás="103">
        <értékelés>3</értékelés>
    </részvétel>
    <részvétel kutya="2" foglalkozás="103">
        <értékelés>3</értékelés>
    </részvétel>
    <részvétel kutya="3" foglalkozás="103">
        <értékelés>5</értékelés>
        <jutalom>Kislabda</jutalom>
    </részvétel>

<!--Kutyatrénerek adatai-->
<tréner tkód="51">
    <név>Horváth Zebulon</név>
    <elérhetőség>
        <email>horvathz@ebiskola.hu</email>
        <telefon>06709186254</telefon>
    </elérhetőség>
    <szakterület>kutyapszichológia</szakterület>
    <órabér>3600</órabér>
</tréner>
<tréner tkód="52">
    <név>Horváth Indigó</név>
    <elérhetőség>
        <email>horvathi@ebiskola.hu</email>
        <telefon>06704567821</telefon>
    </elérhetőség>
    <órabér>2800</órabér>
</tréner>
<tréner tkód="53">
    <név>Polgár Annamária</név>
    <elérhetőség>
        <email>polgar.annam.2@gmail.com</email>
        <email>polgara@ebiskola.hu</email>
        <email>polgara@citromail.hu</email>
        <telefon>+36700670891</telefon>
    </elérhetőség>
    <szakterület>agility</szakterület>
    <órabér>3250</órabér>
</tréner>

</kutyaiskola>

```

1d)

Az XML dokumentum alapján XMLSchema készítése - saját típusok, ref, key, keyref, speciális elemek

Az XML dokumentum validálására XMLSchema sémaleíró dokumentumot hoztam létre.

A név és az elérhetőség elemeket referenciával illesztettem be a megfelelő helyekre. Létrehoztam több saját egyszerű típust, például a kutyák nemének, a tarifa típusának vagy az aktivitás jellemzőnek a megadásához. Az óra típusnál minimum és maximum értékeket jelöltem ki, figyelembe véve azt, hogy a foglalkozások csak 6 és 21 óra között lehetnek megtartva. Az értékelés típus lehetséges értékeit szintén korlátoztam, hogy azok 1 és 5 közé essenek. Az e-mail címekre és a telefonszámokra vonatkozó korlátozások megadásához egyszerű reguláris kifejezéseket használtam. Telefonszámok esetén a +36-tal és a 06-tal kezdődő formátum is megengedett.

Ezt követően komplex típusokat hoztam létre, majd ezeket felhasználva megadtam a kutyaiskola gyökérelém szerkezetét az elsődleges kulcsokkal és idegen kulcsokkal együtt. A tagság-kutya 1:1 kapcsolathoz unique megszorítást vezettem be.

Az így létrejött séma alapján az XML dokumentumot Visual Studio Code fejlesztőkörnyezetben sikeresen validáltam.

Az XMLSchema dokumentum tartalma:

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

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

  <!--Elemek ref-hez-->
  <xs:element name="név" type="xs:string" />
  <xs:element name="elérhetőség" type="elérhetőségTípus" />

  <!-- Saját egyszerű típusok meghatározása, megszorítás -->
  <xs:simpleType name="nemTípus">
    <xs:restriction base="xs:string">
      <xs:enumeration value="szuka" />
      <xs:enumeration value="kan" />
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="tarifaTípus">
    <xs:restriction base="xs:string">
      <xs:enumeration value="normál" />
      <xs:enumeration value="kedvezményes" />
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="aktívTípus">
    <xs:restriction base="xs:string">
      <xs:enumeration value="igen" />
      <xs:enumeration value="nem" />
    </xs:restriction>
  </xs:simpleType>
```

```

<xs:simpleType name="óraTípus">
  <xs:restriction base="xs:nonNegativeInteger">
    <xs:minExclusive value="6" />
    <xs:maxExclusive value="21" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="emailTípus">
  <xs:restriction base="xs:string">
    <xs:pattern value="[a-z]([a-z0-9_.-]*)@[a-z]([a-z0-9_.-]*).[a-z]{2,4}" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="telefonTípus">
  <xs:restriction base="xs:string">
    <xs:pattern value="([+]?)\d{11}" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="értékelésTípus">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="1"/>
    <xs:maxInclusive value="5"/>
  </xs:restriction>
</xs:simpleType>

<!--Komplex típusokhoz saját típus meghatározása, sorrendiség, számosság etc. -->
<xs:complexType name="elérhetőségTípus">
  <xs:sequence>
    <xs:element name="email" type="emailTípus" minOccurs="1" maxOccurs="5" />
    <xs:element name="telefon" type="telefonTípus" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="időpontTípus">
  <xs:sequence>
    <xs:element name="dátum" type="xs:date" />
    <xs:element name="kezdet" type="óraTípus" />
    <xs:element name="vég" type="óraTípus" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="kutyaTípus">
  <xs:sequence>
    <xs:element ref="név" />
    <xs:element name="fajta" type="xs:string" />
    <xs:element name="szül_dátum" type="xs:date" />
    <xs:element name="nem" type="nemTípus" />
  </xs:sequence>
  <xs:attribute name="kkód" type="xs:integer" use="required" />
  <xs:attribute name="gazdi" type="xs:integer" use="required" />
  <xs:attribute name="tagság" type="xs:integer" use="required" />
</xs:complexType>
<xs:complexType name="tagságTípus">
  <xs:all>

```

```

        <xs:element name="kezdőnap" type="xs:date" />
        <xs:element name="tarifa" type="tarifaTípus" />
    </xs:all>
    <xs:attribute name="tagkód" type="xs:integer" use="required" />
    <xs:attribute name="aktív" type="aktívTípus" use="optional" default="igen" />
</xs:complexType>
<xs:complexType name="gazdiTípus">
    <xs:sequence>
        <xs:element ref="név" />
        <xs:element ref="elérhetőség" />
        <xs:element name="szül_év" type="xs:gYear" />
    </xs:sequence>
    <xs:attribute name="gkód" type="xs:integer" use="required" />
</xs:complexType>
<xs:complexType name="foglalkozásTípus">
    <xs:sequence>
        <xs:element name="téma" type="xs:string" minOccurs="1" maxOccurs="unbounded"/>
        <xs:element name="helyszín" type="xs:string" />
        <xs:element name="időpont" type="időpontTípus" />
    </xs:sequence>
    <xs:attribute name="fkód" type="xs:integer" use="required" />
    <xs:attribute name="vezető" type="xs:integer" use="required" />
</xs:complexType>
<xs:complexType name="részvételTípus">
    <xs:sequence>
        <xs:element name="értékelés" type="értékelésTípus" />
        <xs:element name="jutalom" type="xs:string" minOccurs="0" maxOccurs="1" />
    </xs:sequence>
    <xs:attribute name="kutya" type="xs:integer" use="required" />
    <xs:attribute name="foglalkozás" type="xs:integer" use="required" />
</xs:complexType>
<xs:complexType name="trénerTípus">
    <xs:sequence>
        <xs:element ref="név" />
        <xs:element ref="elérhetőség"/>
        <xs:element name="szakterület" type="xs:string" minOccurs="0" maxOccurs="1" />
        <xs:element name="órabér" type="xs:positiveInteger" />
    </xs:sequence>
    <xs:attribute name="tkód" type="xs:integer" use="required" />
</xs:complexType>

<!-- Gyökérelemtől az elemek felhasználása -->
<xs:element name="kutyaiskola">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="kutya" type="kutyaTípus" minOccurs="0" maxOccurs="100" />
            <xs:element name="tagság" type="tagságTípus" minOccurs="0" maxOccurs="100" />
            <xs:element name="gazdi" type="gazdiTípus" minOccurs="0" maxOccurs="100" />
            <xs:element name="foglalkozás" type="foglalkozásTípus" minOccurs="0"
                maxOccurs="unbounded" />
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

```

        <xs:element name="részvétel" type="részvételTípus" minOccurs="0"
            maxOccurs="unbounded" />
        <xs:element name="tréner" type="trénerTípus" minOccurs="1" maxOccurs="100" />
    </xs:sequence>
</xs:complexType>

<!-- Elsődleges kulcsok -->
<xs:key name="kutya_kulcs">
    <xs:selector xpath="kutya" />
    <xs:field xpath="@kkód" />
</xs:key>
<xs:key name="tagság_kulcs">
    <xs:selector xpath="tagság" />
    <xs:field xpath="@tagkód" />
</xs:key>
<xs:key name="gazdi_kulcs">
    <xs:selector xpath="gazdi" />
    <xs:field xpath="@gkód" />
</xs:key>
<xs:key name="foglalkozás_kulcs">
    <xs:selector xpath="foglalkozás" />
    <xs:field xpath="@fkód" />
</xs:key>
<xs:key name="tréner_kulcs">
    <xs:selector xpath="tréner" />
    <xs:field xpath="@tkód" />
</xs:key>

<!-- Idegen kulcsok -->
<xs:keyref name="kutya_tagság_kulcs" refer="tagság_kulcs">
    <xs:selector xpath="kutya" />
    <xs:field xpath="@tagság" />
</xs:keyref>
<xs:keyref name="kutya_gazdi_kulcs" refer="gazdi_kulcs">
    <xs:selector xpath="kutya" />
    <xs:field xpath="@gazdi" />
</xs:keyref>
<xs:keyref name="foglalkozás_tréner_kulcs" refer="tréner_kulcs">
    <xs:selector xpath="foglalkozás" />
    <xs:field xpath="@vezető" />
</xs:keyref>
<xs:keyref name="részvétel_kutya_kulcs" refer="kutya_kulcs">
    <xs:selector xpath="részvétel" />
    <xs:field xpath="@kutya" />
</xs:keyref>
<xs:keyref name="részvétel_foglalkozás_kulcs" refer="foglalkozás_kulcs">
    <xs:selector xpath="részvétel" />
    <xs:field xpath="@foglalkozás" />
</xs:keyref>

```

```
<!-- Az 1:1 kapcsolat megvalósítása -->
<xs:unique name="kutya_tagság_1_1">
  <xs:selector xpath="kutya" />
  <xs:field xpath="@tagság" />
</xs:unique>

</xs:element>

</xs:schema>
```

2. feladat

2a)

Adatolvasás

A DomReadYd11nl osztály beolvassa a kutyaiskola XML dokumentum tartalmát, majd kiírja azt a konzolra és egy fájlba (*XMLReadYd11nl.xml*) strukturált formában. Ennek az osztálynak a public minősítójú metódusait hívja a többi osztály is, amikor XML csomópontot kell kiírniuk vagy XML dokumentumot beolvasniuk. Beolvasás után a DOM fából törölöm az üres (whitespace karaktereket tartalmazó) szöveges csomópontokat, melyek a forrás XML dokumentumban található sortörések miatt alakulnak ki. A fájlba íráshoz transformert használok, míg a konzolra íráskor lépésenként haladok végig a DOM fán, csomópontról csomópontra lépve.

Kétféle strukturált kiírást készítettem a feladathoz. Az egyik a kutyaiskola XML dokumentumra specifikus, az elemeket tag név alapján olvassa be, az egyed típusokat jelölő kommentek előtt üres sort hagy. A másik univerzálisan bármely XML dokumentumra (vagy annak egy részfájára) működik, és tartalmaz színezést is: a konzolon a PI kék, a kommentek pedig zöld színnel jelennek meg.

Output a konzolon:

A beolvasott dokumentum:

```
<?xml version="1.0" encoding="UTF-8" ?>
<kutyaiskola>
  <!--Kutyák, akik be vannak regisztrálva-->
  <kutya gazdi="11" kkód="1" tagság="1000">
    <név>Bizsu</név>
    <fajta>yorkshire terrier</fajta>
    <szül_dátum>2022-10-10</szül_dátum>
    <nem>kan</nem>
  </kutya>
  <kutya gazdi="11" kkód="2" tagság="1002">
    <név>Buksi</név>
    <fajta>németjuhász</fajta>
    <szül_dátum>2021-05-01</szül_dátum>
    <nem>kan</nem>
  </kutya>
  <kutya gazdi="12" kkód="3" tagság="1001">
    <név>Bella</név>
    <fajta>magyar vizsla</fajta>
    <szül_dátum>2023-04-23</szül_dátum>
    <nem>szuka</nem>
  </kutya>
  <kutya gazdi="13" kkód="4" tagság="1003">
    <név>Mister</név>
    <fajta>afgán agár</fajta>
    <szül_dátum>2016-09-15</szül_dátum>
    <nem>kan</nem>
  </kutya>
  <!--Kutyák tagság adatai-->
  <tagság aktív="igen" tagkód="1000">
    <kezdőnap>2023-03-01</kezdőnap>
    <tarifa>kedvezményes</tarifa>
  </tagság>
  <tagság tagkód="1001">
    <kezdőnap>2023-10-01</kezdőnap>
    <tarifa>normál</tarifa>
  </tagság>
```



```

<tagság tagkód="1002">
  <kezdőnap>2022-02-28</kezdőnap>
  <tarifa>normál</tarifa>
</tagság>
<tagság aktív="nem" tagkód="1003">
  <kezdőnap>2022-01-01</kezdőnap>
  <tarifa>kedvezményes</tarifa>
</tagság>
<!--Gazdik adatai-->
<gazdi gkód="11">
  <név>Kiss Katalin</név>
  <elérhetőség>
    <email>kisskat@gmail.com</email>
    <email>kati@kkconsulting.hu</email>
    <email>zoltan.kiss@gmail.com</email>
    <telefon>+36701234987</telefon>
  </elérhetőség>
  <szül_év>1980</szül_év>
</gazdi>
<gazdi gkód="12">
  <név>Kovács Béláné Folt Irén</név>
  <elérhetőség>
    <email>incike@citromail.hu</email>
    <telefon>+36208877665</telefon>
  </elérhetőség>
  <szül_év>1961</szül_év>
</gazdi>
<gazdi gkód="13">
  <név>Petőfi Mór</név>
  <elérhetőség>
    <email>borok.mortol@gmail.com</email>
    <email>talprabaradlay@gmail.com</email>
    <telefon>+36506060768</telefon>
  </elérhetőség>
  <szül_év>1972</szül_év>
</gazdi>
<!--Foglalkozások-->
<foglalkozás fkód="100" vezető="51">
  <téma>elsősegély oktatás</téma>
  <téma>kutyapszichológia előadás</téma>
  <téma>engedelmesség tréning</téma>
  <helyszín>gyakorlóaréna</helyszín>
  <időpont>
    <dátum>2023-10-01</dátum>
    <kezdet>14</kezdet>
    <vég>18</vég>
  </időpont>
</foglalkozás>
<foglalkozás fkód="101" vezető="53">
  <téma>agility foglalkozás</téma>
  <téma>activity foglalkozás</téma>
  <helyszín>akadálypálya</helyszín>
  <időpont>
    <dátum>2023-10-08</dátum>
    <kezdet>15</kezdet>
    <vég>16</vég>
  </időpont>
</foglalkozás>
<foglalkozás fkód="102" vezető="51">
  <téma>engedelmesség tréning</téma>
  <helyszín>akadálypálya</helyszín>
  <időpont>
    <dátum>2023-10-15</dátum>

```

```

        <kezdet>15</kezdet>
        <vég>17</vég>
    </időpont>
</foglalkozás>
<foglalkozás fkód="103" vezető="51">
    <téma>engedelmesség tréning</téma>
    <helyszín>akadálypálya</helyszín>
    <időpont>
        <dátum>2023-11-02</dátum>
        <kezdet>10</kezdet>
        <vég>12</vég>
    </időpont>
</foglalkozás>
<!--Kutya-foglalkozás kapcsoló elemek avagy részvételi adatok-->
<részvétel foglalkozás="100" kutya="1">
    <értékelés>4</értékelés>
</részvétel>
<részvétel foglalkozás="100" kutya="2">
    <értékelés>5</értékelés>
    <jutalom>Rágócsont</jutalom>
</részvétel>
<részvétel foglalkozás="101" kutya="2">
    <értékelés>5</értékelés>
    <jutalom>Purina száraz kutyatáp</jutalom>
</részvétel>
<részvétel foglalkozás="101" kutya="3">
    <értékelés>2</értékelés>
    <jutalom>Rágócsont</jutalom>
</részvétel>
<részvétel foglalkozás="102" kutya="1">
    <értékelés>5</értékelés>
    <jutalom>Kislabda</jutalom>
</részvétel>
<részvétel foglalkozás="102" kutya="2">
    <értékelés>5</értékelés>
    <jutalom>Játszókötélt</jutalom>
</részvétel>
<részvétel foglalkozás="103" kutya="1">
    <értékelés>3</értékelés>
</részvétel>
<részvétel foglalkozás="103" kutya="2">
    <értékelés>3</értékelés>
</részvétel>
<részvétel foglalkozás="103" kutya="3">
    <értékelés>5</értékelés>
    <jutalom>Kislabda</jutalom>
</részvétel>
<!--Kutyatrénerek adatai-->
<tréner tkód="51">
    <név>Horváth Zebulon</név>
    <elérhetőség>
        <email>horvathz@ebiskola.hu</email>
        <telefon>06709186254</telefon>
    </elérhetőség>
    <szakterület>kutyapszichológia</szakterület>
    <órabér>3600</órabér>
</tréner>
<tréner tkód="52">
    <név>Horváth Indigó</név>
    <elérhetőség>
        <email>horvathi@ebiskola.hu</email>
        <telefon>06704567821</telefon>
    </elérhetőség>

```

```

        <órabér>2800</órabér>
    </tréner>
    <tréner tkód="53">
        <név>Polgár Annamária</név>
        <elérhetőség>
            <email>polgar.annam.2@gmail.com</email>
            <email>polgara@ebiskola.hu</email>
            <email>polgara@citromail.hu</email>
            <telefon>+36700670891</telefon>
        </elérhetőség>
        <szakterület>agility</szakterület>
        <órabér>3250</órabér>
    </tréner>
</kutyaiskola>

```

Az osztály kódja:

```

package hu.domparse.yd11nl;

import java.io.File;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.transform.OutputKeys;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerException;
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;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;

public class DomReadYd11nl {
    public static void main(String[] args) {
        try {
            //Output XML file megadása
            File newXmlFile = new File("XMLReadYd11nl.xml");
            StreamResult newXmlStream = new StreamResult(newXmlFile);
            //XML dokumentum beolvasása
            Document document = parseXML("XMLYd11nl.xml");
            //Kiírás az output fájlba
            writeDocument(document, newXmlStream);
            //Kiírás konzolra általános XML kiíróval
            System.out.println("A beolvasott dokumentum:\n\n"+formatXML(document));

```

```

        //Kiírás konzolra feladatspecifikus kiíróval
        //System.out.println("A beolvasott
        dokumentum:\n\n"+getStructuredDocument(document));
    } catch (IOException | SAXException | ParserConfigurationException |
    TransformerException e) {
        e.printStackTrace();
    }
}

//Adott nevű XML dokumentumból Document készítése
public static Document parseXML(String fileName) throws SAXException, IOException,
ParserConfigurationException {
    File xmlFile = new File(fileName);
    DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
    DocumentBuilder builder = factory.newDocumentBuilder();
    Document document = builder.parse(xmlFile);
    Node root = document.getDocumentElement();
    root.normalize();
    cleanDocument(root);
    return document;
}

//Üres node-ok törlése a dokumentumból (sortörések miatti)
private static void cleanDocument(Node root) {
    NodeList nodes = root.getChildNodes();
    List<Node> toDelete = new ArrayList<>();
    for(int i=0; i<nodes.getLength(); i++) {
        if(nodes.item(i).getNodeType()==Node.TEXT_NODE &&
        nodes.item(i).getTextContent().strip().equals("")) {
            toDelete.add(nodes.item(i));
        }else {
            cleanDocument(nodes.item(i));
        }
    }
    for(Node node: toDelete) {
        root.removeChild(node);
    }
}

//DOM fa tartalmának kiírása megadott streambe transformerrel
public static void writeDocument(Document document, StreamResult output) throws
TransformerException {
    TransformerFactory transformerFactory = TransformerFactory.newInstance();
    Transformer transformer;
    transformer = transformerFactory.newTransformer();
    transformer.setOutputProperty(OutputKeys.ENCODING, "UTF-8");
    transformer.setOutputProperty(OutputKeys.INDENT, "yes");

    DOMSource source = new DOMSource(document);
    transformer.transform(source, output);
}

```

```

}

//Kutyaiskola dokumentum strukturált tartalmának visszaadása (feladat specifikus beolvasó)
public static String getStructuredDocument(Document document) {
    //PI
    String output = "<?xml version=\"" + document.getXmlVersion()
    + "\" encoding=\"" + document.getXmlEncoding() + "\"?>\n";
    //Gyökérelem és gyerekelemei
    output += "<" + document.getDocumentElement().getNodeName() + ">\n";
    output += getStructuredDogs(document, 1);
    output += getStructuredMemberships(document, 1);
    output += getStructuredOwners(document, 1);
    output += getStructuredTrainings(document, 1);
    output += getStructuredParticipations(document, 1);
    output += getStructuredTrainers(document, 1);
    output += "\n</" + document.getDocumentElement().getNodeName() + ">";

    return output;
}

//Strukturált kutya elemek
private static String getStructuredDogs(Document document, int indent) {
    String output = "";
    //Végigiterálunk az összes kutyán
    NodeList dogNodes = document.getElementsByTagName("kutya");
    for(int i=0; i<dogNodes.getLength(); i++) {
        Element dog = (Element) dogNodes.item(i);
        //Ha van az elem előtt comment, kiírjuk
        output += getComment(dog.getPreviousSibling(), indent);
        //Attribútumok
        String id = dog.getAttribute("kkód");
        String owner = dog.getAttribute("gazdi");
        String membership = dog.getAttribute("tagság");
        //Gyerekelemek
        Element name = (Element) dog.getElementsByTagName("név").item(0);
        Element breed = (Element) dog.getElementsByTagName("fajta").item(0);
        Element birthDate = (Element) dog.getElementsByTagName("szül_dátum").item(0);
        Element gender = (Element) dog.getElementsByTagName("nem").item(0);
        //Strukturált Stringbe írás
        output += indent(indent) + "<kutya kkód=\"" + id + "\" gazdi=\"" + owner
        + "\" tagság=\"" + membership + "\">\n";
        indent++;
        output += indent(indent)
        + getStructuredSimpleElement("név", name.getTextContent());
        output += indent(indent)
        + getStructuredSimpleElement("fajta", breed.getTextContent());
        output += indent(indent)
        + getStructuredSimpleElement("szül_dátum", birthDate.getTextContent());
        output += indent(indent)
        + getStructuredSimpleElement("nem", gender.getTextContent());
    }
}

```

```

        indent--;
        output += indent(indent)+"</kutya>\n";
    }
    return output;
}

//Strukturált tagság elemek
private static String getStructuredMemberships(Document document, int indent) {
    String output = "";
    NodeList membershipNodes = document.getElementsByTagName("tagság");
    for(int i=0; i<membershipNodes.getLength(); i++) {
        Element membership = (Element) membershipNodes.item(i);
        //Ha van az elem előtt comment, kiírjuk
        output += getComment(membership.getPreviousSibling(), indent);
        //Attribútumok
        String id = membership.getAttribute("tagkód");
        String active = membership.getAttribute("aktív");
        //Ha van aktív attribútum, kiírjuk, egyébként üres String helyette
        String activity = active.isEmpty() ? "" : " aktív=\""+active+"\"";
        //Gyerekelemek
        Element firstDay = (Element) membership.getElementsByTagName("kezdőnap").item(0);
        Element rate = (Element) membership.getElementsByTagName("tarifa").item(0);
        //Strukturált Stringbe írás
        output += indent(indent)+"<tagság tagkód=\""+id+activity+">\n";
        indent++;
        output += indent(indent)
            +getStructuredSimpleElement("kezdőnap", firstDay.getTextContent());
        output += indent(indent)
            +getStructuredSimpleElement("tarifa", rate.getTextContent());
        indent--;
        output += indent(indent)+"</tagság>\n";
    }
    return output;
}

//Strukturált gazdi elemek
private static String getStructuredOwners(Document document, int indent) {
    String output = "";
    NodeList ownerNodes = document.getElementsByTagName("gazdi");
    for(int i=0; i<ownerNodes.getLength(); i++) {
        Element owner = (Element) ownerNodes.item(i);
        //Ha van az elem előtt comment, kiírjuk
        output += getComment(owner.getPreviousSibling(), indent);
        //Attribútumok
        String id = owner.getAttribute("gkód");
        //Gyerekelemek
        Element name = (Element) owner.getElementsByTagName("név").item(0);
        Element contact = (Element) owner.getElementsByTagName("elérhetőség").item(0);
        Element birthYear = (Element) owner.getElementsByTagName("szül_év").item(0);
        //Strukturált Stringbe írás

```

```

        output += indent(indent)+"<gazdi gkód=\""+id+"\">\n";
        indent++;
        output += indent(indent)
        +getStructuredSimpleElement("név", name.getTextContent());
        output += getStructuredContact(contact, indent);
        output += indent(indent)
        +getStructuredSimpleElement("szül_év", birthYear.getTextContent());
        indent--;
        output += indent(indent)+"</gazdi>\n";
    }
    return output;
}

//Strukturált foglalkozás elemek
private static String getStructuredTrainings(Document document, int indent) {
    String output = "";
    NodeList trainingNodes = document.getElementsByTagName("foglalkozás");
    for(int i=0; i<trainingNodes.getLength(); i++) {
        Element training = (Element) trainingNodes.item(i);
        //Ha van az elem előtt comment, kiírjuk
        output += getComment(training.getPreviousSibling(), indent);
        //Attribútumok
        String id = training.getAttribute("fkód");
        String trainer = training.getAttribute("vezető");
        //Gyerekelemek
        NodeList topics = training.getElementsByTagName("téma");
        Element place = (Element) training.getElementsByTagName("helyszín").item(0);
        Element time = (Element) training.getElementsByTagName("időpont").item(0);
        Element day = (Element) time.getElementsByTagName("dátum").item(0);
        Element begin = (Element) time.getElementsByTagName("kezdet").item(0);
        Element end = (Element) time.getElementsByTagName("vég").item(0);
        //Strukturált Stringbe írás
        output += indent(indent)+"<foglalkozás fkód=\""+id
        +"\" vezető=\""+trainer+"\">\n";
        indent++;
        //A téma többszörös elem, mindegyik előfordulását kiírjuk
        for(int j=0; j<topics.getLength(); j++) {
            output += indent(indent)+getStructuredSimpleElement("téma",
            ((Element)topics.item(j)).getTextContent());
        }
        output += indent(indent)
        +getStructuredSimpleElement("helyszín", place.getTextContent());
        //Az időpont gyerekelem kiírása elemtartalmával együtt
        output += indent(indent)+"<időpont>\n";
        indent++;
        output += indent(indent)
        +getStructuredSimpleElement("dátum", day.getTextContent());
        output += indent(indent)
        +getStructuredSimpleElement("kezdet", begin.getTextContent());
        output += indent(indent)+getStructuredSimpleElement("vég", end.getTextContent());
    }
}

```

```

        indent--;
        output += indent(indent)+"</időpont>\n";
        indent--;
        output += indent(indent)+"</foglalkozás>\n";
    }
    return output;
}

//Strukturált részvétel elemek
private static String getStructuredParticipations(Document document, int indent) {
    String output = "";
    NodeList participationNodes = document.getElementsByTagName("részvétel");
    for(int i=0; i<participationNodes.getLength(); i++) {
        Element participation = (Element) participationNodes.item(i);
        //Ha van az elem előtt comment, kiírjuk
        output += getComment(participation.getPreviousSibling(), indent);
        //Attribútumok
        String dog = participation.getAttribute("kutya");
        String training = participation.getAttribute("foglalkozás");
        //Gyerekelemek
        Element evaluation =
            (Element) participation.getElementsByTagName("értékelés").item(0);
        NodeList prize = participation.getElementsByTagName("jutalom");
        //Strukturált Stringbe írás
        output += indent(indent)+"<részvétel kutya=\""+dog
            +"\" foglalkozás=\""+training+"\">\n";
        indent++;
        output += indent(indent)
            +getStructuredSimpleElement("értékelés", evaluation.getTextContent());
        //A jutalom elem opcionális, csak akkor írjuk ki, ha létezik
        output += prize.getLength()==0 ? "" : indent(indent)
            +getStructuredSimpleElement("jutalom",
                ((Element)prize.item(0)).getTextContent());
        indent--;
        output += indent(indent)+"</részvétel>\n";
    }
    return output;
}

//Strukturált tréner elemek
private static String getStructuredTrainers(Document document, int indent) {
    String output = "";
    NodeList trainerNodes = document.getElementsByTagName("tréner");
    for(int i=0; i<trainerNodes.getLength(); i++) {
        Element trainer = (Element) trainerNodes.item(i);
        //Ha van az elem előtt comment, kiírjuk
        output += getComment(trainer.getPreviousSibling(), indent);
        //Attribútumok
        String id = trainer.getAttribute("tkód");
        //Gyerekelemek

```



```

        Element name = (Element) trainer.getElementsByTagName("név").item(0);
        Element contact = (Element) trainer.getElementsByTagName("elérhetőség").item(0);
        NodeList expertise = trainer.getElementsByTagName("szakterület");
        Element wage = (Element) trainer.getElementsByTagName("órabér").item(0);
        //Strukturált Stringbe írás
        output += indent(indent)+"<tréner tkód=\""+id+"\">\n";
        indent++;
        output += indent(indent)
        +getStructuredSimpleElement("név", name.getTextContent());
        output += getStructuredContact(contact, indent);
        //A szakterület elem opcionális, csak akkor írjuk ki, ha létezik
        output += expertise.getLength() == 0 ? "" : indent(indent)
        +getStructuredSimpleElement("szakterület", expertise.item(0).getTextContent());
        output += indent(indent)
        +getStructuredSimpleElement("órabér", wage.getTextContent());
        indent--;
        output += indent(indent)+"</tréner>\n";
    }
    return output;
}

//Strukturált elérhetőség elem
private static String getStructuredContact(Element contact, int indent) {
    String output = "";
    //Gyerekelemek
    NodeList emails = contact.getElementsByTagName("email");
    Element phone = (Element)contact.getElementsByTagName("telefon").item(0);
    //Strukturált Stringbe írás
    output += indent(indent)+"<elérhetőség>\n";
    indent++;
    //Az email többszörös elem, minden előfordulását kiírjuk
    for(int i=0; i<emails.getLength(); i++) {
        output+=indent(indent)+getStructuredSimpleElement("email",
            ((Element)emails.item(i)).getTextContent());
    }
    output += indent(indent)
    +getStructuredSimpleElement("telefon", phone.getTextContent());
    indent--;
    output += indent(indent)+"</elérhetőség>\n";
    return output;
}

//Szöveges tartalmú elem visszaadása strukturált formában
private static String getStructuredSimpleElement(String tag, String content) {
    return "<"+tag+">"+content+"</"+tag+">\n";
}

//Komment csomópont Stringbe írása strukturáltan
private static String getComment(Node examinedNode, int indent) {
    String output = "";

```

```

        if(examinedNode.getNodeType()==Node.COMMENT_NODE) {
            output+="\n"+indent(indent)+"<!--"+((Comment)examinedNode).getData()+"-->\n";
        }
        return output;
    }

    //DOM fa visszaadása XML-ben formázva (általános XML beolvasó, színezett PI és kommentek)
    public static String formatXML(Document document) {
        //Prolog kiírva kék színnel, utána vissza feketére
        String pi = "\u001B[34m<?xml version=\""+document.getXmlVersion()
            +"\\" encoding=\""+document.getXmlEncoding()+"\\" ?>\u001B[30m";
        return pi+formatElement(document.getDocumentElement(), 0);
    }

    //Adott XML csomópontnak és tartalmának strukturált Stringgé konvertálása formázott
    kiíráshoz (általános XML beolvasó)
    public static String formatElement(Node node, int indent) {
        //Ha node nem elem, üres Stringgel térünk vissza
        if (node.getNodeType() != Node.ELEMENT_NODE) {
            return "";
        }
        //Egyébként felépítjük az xml elemet
        String output = "\n";
        output += indent(indent)+"<" + ((Element) node).getTagName();
        //Attribútumok formázott felvétele, ha vannak
        if (node.hasAttributes()) {
            for (int i = 0; i < node.getAttributes().getLength(); i++) {
                Node attribute = node.getAttributes().item(i);
                output += " " + attribute.getNodeName() + "="
                    + attribute.getNodeValue() + "\"";
            }
        }
        output += ">";
        //Gyerekelemek feldolgozása
        NodeList children = node.getChildNodes();
        for (int i = 0; i < children.getLength(); i++) {
            //Szöveges tartalom
            if(children.item(i).getNodeType()==Node.TEXT_NODE)
                return output+=node.getTextContent()
                    + "</" + ((Element) node).getTagName() + ">";
            //Gyerekelem
            if(children.item(i).getNodeType()==Node.ELEMENT_NODE)
                output+=formatElement(children.item(i), indent+1);
            //Komment (zöld színnel)
            if(children.item(i).getNodeType()==Node.COMMENT_NODE)
                output+="\n"+indent(indent+1)+"\u001B[32m<!--"
                    +((Comment)children.item(i)).getData()+"-->\u001B[30m";
        }
        output+="\n"+indent(indent)+"</" + ((Element) node).getTagName() + ">";
    }

```

```
        return output;
    }

    //Tabulálás
    private static String indent(int indent) {
        return " ".repeat(indent);
    }
}
```

2b)

Adatmódosítás

A DomModifyYd1nl osztály meghatározott módosításokat hajt végre a beolvasott DOM fában, ezeknél egyenként kiírja az előtte-utána állapotokat, valamint az eredményül kapott módosult DOM fát is kiírja a konzolra a futás végén.

A végrehajtott 5 módosítás:

1. Az akadálypálya helyszín átírása minden előfordulásnál I. pályára
2. A Bizsu nevű kutya tagságának passzívra állítása
3. Kovács Béláné Folt Irén nevének átírása Folt Irénre, valamint születési évének átírása 1959-re
4. A 3. tréner 2. email címének törlése
5. Új foglalkozás felvétele automatikusan generált kulccsal, melynek értéke eggyel nagyobb, mint az eddigi fkód értékek maximuma

Output a konzolon:

1. módosítás

ELŐTTE:

```
<foglalkozás fkód="100" vezető="51">
  <téma>elsősegély oktatás</téma>
  <téma>kutyapszichológia előadás</téma>
  <téma>engedelmesség tréning</téma>
  <helyszín>gyakorlóaréna</helyszín>
  <időpont>
    <dátum>2023-10-01</dátum>
    <kezdet>14</kezdet>
    <vég>18</vég>
  </időpont>
</foglalkozás>
<foglalkozás fkód="101" vezető="53">
  <téma>agility foglalkozás</téma>
  <téma>activity foglalkozás</téma>
  <helyszín>akadálypálya</helyszín>
  <időpont>
    <dátum>2023-10-08</dátum>
    <kezdet>15</kezdet>
    <vég>16</vég>
  </időpont>
</foglalkozás>
<foglalkozás fkód="102" vezető="51">
  <téma>engedelmesség tréning</téma>
  <helyszín>akadálypálya</helyszín>
  <időpont>
    <dátum>2023-10-15</dátum>
    <kezdet>15</kezdet>
    <vég>17</vég>
  </időpont>
</foglalkozás>
<foglalkozás fkód="103" vezető="51">
  <téma>engedelmesség tréning</téma>
  <helyszín>akadálypálya</helyszín>
  <időpont>
    <dátum>2023-11-02</dátum>
    <kezdet>10</kezdet>
    <vég>12</vég>
  </időpont>
</foglalkozás>
```

```

</foglalkozás>
UTÁNA:
<foglalkozás fkód="100" vezető="51">
  <téma>elsősegély oktatás</téma>
  <téma>kutyapszichológia előadás</téma>
  <téma>engedelmesség tréning</téma>
  <helyszín>gyakorlóaréna</helyszín>
  <időpont>
    <dátum>2023-10-01</dátum>
    <kezdet>14</kezdet>
    <vég>18</vég>
  </időpont>
</foglalkozás>
<foglalkozás fkód="101" vezető="53">
  <téma>agility foglalkozás</téma>
  <téma>activity foglalkozás</téma>
  <!--Átírt helyszín-->
  <helyszín>I. pálya</helyszín>
  <időpont>
    <dátum>2023-10-08</dátum>
    <kezdet>15</kezdet>
    <vég>16</vég>
  </időpont>
</foglalkozás>
<foglalkozás fkód="102" vezető="51">
  <téma>engedelmesség tréning</téma>
  <!--Átírt helyszín-->
  <helyszín>I. pálya</helyszín>
  <időpont>
    <dátum>2023-10-15</dátum>
    <kezdet>15</kezdet>
    <vég>17</vég>
  </időpont>
</foglalkozás>
<foglalkozás fkód="103" vezető="51">
  <téma>engedelmesség tréning</téma>
  <!--Átírt helyszín-->
  <helyszín>I. pálya</helyszín>
  <időpont>
    <dátum>2023-11-02</dátum>
    <kezdet>10</kezdet>
    <vég>12</vég>
  </időpont>
</foglalkozás>

```

2. módosítás

```

ELŐTTE:
<tagság aktív="igen" tagkód="1000">
  <kezdőnap>2023-03-01</kezdőnap>
  <tarifa>kedvezményes</tarifa>
</tagság>
UTÁNA:
<tagság aktív="nem" tagkód="1000">
  <kezdőnap>2023-03-01</kezdőnap>
  <tarifa>kedvezményes</tarifa>
  <!--Passzivált kutya-->
</tagság>

```

3. módosítás

```

ELŐTTE:
<gazdi gkód="12">
  <név>Kovács Béláné Folt Irén</név>
  <elérhetőség>

```

```

        <email>incike@citromail.hu</email>
        <telefon>+36208877665</telefon>
    </elérhetőség>
    <szül_év>1961</szül_év>
</gazdi>
UTÁNA:
<gazdi gkód="12">
    <!--Átírt név és születési év-->
    <név>Folt Irén</név>
    <elérhetőség>
        <email>incike@citromail.hu</email>
        <telefon>+36208877665</telefon>
    </elérhetőség>
    <szül_év>1959</szül_év>
</gazdi>

```

4. módosítás

ELŐTTE:

```

<tréner tkód="53">
    <név>Polgár Annamária</név>
    <elérhetőség>
        <email>polgar.annam.2@gmail.com</email>
        <email>polgara@ebiskola.hu</email>
        <email>polgara@citromail.hu</email>
        <telefon>+36700670891</telefon>
    </elérhetőség>
    <szakterület>agility</szakterület>
    <órabér>3250</órabér>
</tréner>
UTÁNA:
<tréner tkód="53">
    <név>Polgár Annamária</név>
    <elérhetőség>
        <email>polgar.annam.2@gmail.com</email>
        <!--Törölt email helye-->
        <email>polgara@citromail.hu</email>
        <telefon>+36700670891</telefon>
    </elérhetőség>
    <szakterület>agility</szakterület>
    <órabér>3250</órabér>
</tréner>

```

5. módosítás

ÚJ FOGLALKOZÁS:

```

<foglalkozás fkód="104" vezető="52">
    <téma>engedelmesség tréning</téma>
    <téma>kutyatulajdonos alapismeretek</téma>
    <helyszín>hátsó udvar</helyszín>
    <időpont>
        <dátum>2023-11-10</dátum>
        <kezdet>17</kezdet>
        <vég>19</vég>
    </időpont>
</foglalkozás>

```

A módosított dokumentum:

```

<?xml version="1.0" encoding="UTF-8" ?>
<kutyaiskola>
    <!--Kutyák, akik be vannak regisztrálva-->
    <kutya gazdi="11" kkód="1" tagság="1000">
        <név>Bizsu</név>
        <fajta>yorkshire terrier</fajta>
        <szül_dátum>2022-10-10</szül_dátum>
    </kutya>

```

```

    <nem>kan</nem>
</kutya>
<kutya gazdi="11" kkód="2" tagság="1002">
    <név>Buxsi</név>
    <fajta>németjuhász</fajta>
    <szül_dátum>2021-05-01</szül_dátum>
    <nem>kan</nem>
</kutya>
<kutya gazdi="12" kkód="3" tagság="1001">
    <név>Bella</név>
    <fajta>magyar vizsla</fajta>
    <szül_dátum>2023-04-23</szül_dátum>
    <nem>szuka</nem>
</kutya>
<kutya gazdi="13" kkód="4" tagság="1003">
    <név>Mister</név>
    <fajta>afgán agár</fajta>
    <szül_dátum>2016-09-15</szül_dátum>
    <nem>kan</nem>
</kutya>
<!--Kutyák tagság adatai-->
<tagság aktív="nem" tagkód="1000">
    <kezdőnap>2023-03-01</kezdőnap>
    <tarifa>kedvezményes</tarifa>
    <!--Passzivált kutya-->
</tagság>
<tagság tagkód="1001">
    <kezdőnap>2023-10-01</kezdőnap>
    <tarifa>normál</tarifa>
</tagság>
<tagság tagkód="1002">
    <kezdőnap>2022-02-28</kezdőnap>
    <tarifa>normál</tarifa>
</tagság>
<tagság aktív="nem" tagkód="1003">
    <kezdőnap>2022-01-01</kezdőnap>
    <tarifa>kedvezményes</tarifa>
</tagság>
<!--Gazdik adatai-->
<gazdi gkód="11">
    <név>Kiss Katalin</név>
    <elérhetőség>
        <email>kisskat@gmail.com</email>
        <email>kati@kkconsulting.hu</email>
        <email>zoltan.kiss@gmail.com</email>
        <telefon>+36701234987</telefon>
    </elérhetőség>
    <szül_év>1980</szül_év>
</gazdi>
<gazdi gkód="12">
    <!--Átírt név és születési év-->
    <név>Folt Irén</név>
    <elérhetőség>
        <email>incike@citromail.hu</email>
        <telefon>+36208877665</telefon>
    </elérhetőség>
    <szül_év>1959</szül_év>
</gazdi>
<gazdi gkód="13">
    <név>Petőfi Mór</név>
    <elérhetőség>
        <email>borok.mortol@gmail.com</email>
        <email>talprabaradlay@gmail.com</email>

```

```

        <telefon>+36506060768</telefon>
    </elérhetőség>
    <szül_év>1972</szül_év>
</gazdi>
<!--Foglalkozások-->
<foglalkozás fkód="100" vezető="51">
    <téma>elsősegély oktatás</téma>
    <téma>kutyapszichológia előadás</téma>
    <téma>engedelmesség tréning</téma>
    <helyszín>gyakorlóaréna</helyszín>
    <időpont>
        <dátum>2023-10-01</dátum>
        <kezdet>14</kezdet>
        <vég>18</vég>
    </időpont>
</foglalkozás>
<foglalkozás fkód="101" vezető="53">
    <téma>agility foglalkozás</téma>
    <téma>activity foglalkozás</téma>
    <!--Átírt helyszín-->
    <helyszín>I. pálya</helyszín>
    <időpont>
        <dátum>2023-10-08</dátum>
        <kezdet>15</kezdet>
        <vég>16</vég>
    </időpont>
</foglalkozás>
<foglalkozás fkód="102" vezető="51">
    <téma>engedelmesség tréning</téma>
    <!--Átírt helyszín-->
    <helyszín>I. pálya</helyszín>
    <időpont>
        <dátum>2023-10-15</dátum>
        <kezdet>15</kezdet>
        <vég>17</vég>
    </időpont>
</foglalkozás>
<foglalkozás fkód="103" vezető="51">
    <téma>engedelmesség tréning</téma>
    <!--Átírt helyszín-->
    <helyszín>I. pálya</helyszín>
    <időpont>
        <dátum>2023-11-02</dátum>
        <kezdet>10</kezdet>
        <vég>12</vég>
    </időpont>
</foglalkozás>
<!--Új foglalkozás-->
<foglalkozás fkód="104" vezető="52">
    <téma>engedelmesség tréning</téma>
    <téma>kutyatulajdonos alapismeretek</téma>
    <helyszín>hátsó udvar</helyszín>
    <időpont>
        <dátum>2023-11-10</dátum>
        <kezdet>17</kezdet>
        <vég>19</vég>
    </időpont>
</foglalkozás>
<!--Kutya-foglalkozás kapcsoló elemek avagy részvételi adatok-->
<részvétel foglalkozás="100" kutya="1">
    <értékelés>4</értékelés>
</részvétel>
<részvétel foglalkozás="100" kutya="2">

```



```

        <értékelés>5</értékelés>
        <jutalom>Rágócsont</jutalom>
    </részvétel>
    <részvétel foglalkozás="101" kutya="2">
        <értékelés>5</értékelés>
        <jutalom>Purina száraz kutyatáp</jutalom>
    </részvétel>
    <részvétel foglalkozás="101" kutya="3">
        <értékelés>2</értékelés>
        <jutalom>Rágócsont</jutalom>
    </részvétel>
    <részvétel foglalkozás="102" kutya="1">
        <értékelés>5</értékelés>
        <jutalom>Kislabda</jutalom>
    </részvétel>
    <részvétel foglalkozás="102" kutya="2">
        <értékelés>5</értékelés>
        <jutalom>Játszókötélt</jutalom>
    </részvétel>
    <részvétel foglalkozás="103" kutya="1">
        <értékelés>3</értékelés>
    </részvétel>
    <részvétel foglalkozás="103" kutya="2">
        <értékelés>3</értékelés>
    </részvétel>
    <részvétel foglalkozás="103" kutya="3">
        <értékelés>5</értékelés>
        <jutalom>Kislabda</jutalom>
    </részvétel>
    <!--Kutyatrénerek adatai-->
    <tréner tkód="51">
        <név>Horváth Zebulon</név>
        <elérhetőség>
            <email>horvathz@ebiskola.hu</email>
            <telefon>06709186254</telefon>
        </elérhetőség>
        <szakterület>kutyapszichológia</szakterület>
        <órabér>3600</órabér>
    </tréner>
    <tréner tkód="52">
        <név>Horváth Indigó</név>
        <elérhetőség>
            <email>horvathi@ebiskola.hu</email>
            <telefon>06704567821</telefon>
        </elérhetőség>
        <órabér>2800</órabér>
    </tréner>
    <tréner tkód="53">
        <név>Polgár Annamária</név>
        <elérhetőség>
            <email>polgar.annam.2@gmail.com</email>
            <!--Törölt email helye-->
            <email>polgara@citromail.hu</email>
            <telefon>+36700670891</telefon>
        </elérhetőség>
        <szakterület>agility</szakterület>
        <órabér>3250</órabér>
    </tréner>
</kutyaiskola>

```

Az osztály kódja:

```
package hu.domparse.yd11nl;

import java.io.IOException;
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 DomModifyYd11nl {
    public static void main(String[] args) {
        try {
            Document document = DomReadYd11nl.parseXML("XMLYd11nl.xml");

            //1. Az akadálypálya helyszín átírása I. pályára
            System.out.println("\u001B[31m1. módosítás\u001B[30m");
            modifyTrainingPlace(document, "akadálypálya", "I. pálya");
            //2. A Bizsu nevű kutya tagságának passzívra állítása.
            System.out.println("\n\u001B[31m2. módosítás\u001B[30m");
            setToPassive(document, "Bizsu");
            //3. Kovács Béláné Folt Irén átírása Folt Irénre és születési évének átírása
            //1959-re
            System.out.println("\n\u001B[31m3. módosítás\u001B[30m");
            modifyOwnerNameAndBirth(document, "Kovács Béláné Folt Irén", "Folt Irén",
                "1959");
            //4. A 3. tréner 2. email címének törlése.
            System.out.println("\n\u001B[31m4. módosítás\u001B[30m");
            deleteTrainerEmail(document, 3, 2);
            //5. Új foglalkozás felvétele automatikusan generált kóddal (max(fkód)+1)
            System.out.println("\n\u001B[31m5. módosítás\u001B[30m");
            insertTraining(document, "52", new String[] {"engedelmesség tréning",
                "kutyatulajdonos alapismeretek"}, "hátsó udvar", "2023-11-10", "17", "19");

            System.out.println("\n\u001B[31mA módosított dokumentum:\u001B[31m");
            System.out.println(DomReadYd11nl.formatXML(document));

        } catch (IOException | SAXException | ParserConfigurationException e) {
            System.out.println(e.getMessage());
        }
    }

    //Egy foglalkozás helyszín nevének átírása minden előfordulásnál
    private static void modifyTrainingPlace(Document document, String oldPlaceName, String
        newPlaceName) {
        //Az összes adott nevű helyszínnel rendelkező foglalkozás megkeresése
        Element root = document.getDocumentElement();
        NodeList trainings = root.getElementsByTagName("foglalkozás");
```

```

System.out.println("ELŐTTE:");
for(int i=0; i<trainings.getLength(); i++) {
    Element training = (Element)trainings.item(i);
    //Eredeti foglalkozások kiírása
    System.out.println(DomReadYd11nl.formatElement(training, 0).stripLeading());
    Element place = (Element)training.getElementsByTagName("helyszín").item(0);
    if(place.getTextContent().equals(oldPlaceName)) {
        //A helyszínek átírása
        place.setTextContent(newPlaceName);
        training.insertBefore(document.createComment("Átírt helyszín"), place);
    }
}
//A módosítás hatásának kiírása
System.out.println("UTÁNA:");
for(int i=0; i<trainings.getLength(); i++) {
    Element training = (Element)trainings.item(i);
    System.out.println(DomReadYd11nl.formatElement(training, 0).stripLeading());
}

}

//Egy adott nevű kutya státuszának passziválása
private static void setToPassive(Document document, String dogName) {
    //A dogName nevű kutya tagságának kódját lekérjük
    String membershipId = "";
    Element root = document.getDocumentElement();
    NodeList dogs = root.getElementsByTagName("kutya");
    for(int i=0; i<dogs.getLength(); i++) {
        Element dog = (Element)dogs.item(i);
        Element name = (Element)dog.getElementsByTagName("név").item(0);
        if(name.getTextContent().equals(dogName)) {
            membershipId = dog.getAttribute("tagság");
        }
    }
    //Az adott kódú tagságot passziváljuk és kiírjuk az előtte-utána összehasonlítást
    NodeList memberships = root.getElementsByTagName("tagság");
    for(int i=0; i<memberships.getLength(); i++) {
        Element membership = (Element)memberships.item(i);
        if(membership.getAttribute("tagkód").equals(membershipId)) {
            System.out.println("ELŐTTE:"+DomReadYd11nl.formatElement(membership, 0));
            membership.setAttribute("aktív", "nem");
            membership.appendChild(document.createComment("Passzivált kutya"));
            System.out.println("UTÁNA:"+DomReadYd11nl.formatElement(membership, 0));
        }
    }
}

//Egy adott nevű gazdi nevének és születési évének megváltoztatása
private static void modifyOwnerNameAndBirth(Document document, String oldName, String
newName, String newBirthYear) {

```

```

//Végigmegyünk az összes gazdin, akinek a neve oldName, azt newName-re változtatjuk
Element root = document.getDocumentElement();
NodeList owners = root.getElementsByTagName("gazdi");
for(int i=0; i<owners.getLength(); i++) {
    Element owner = (Element)owners.item(i);
    Element name = (Element)owner.getElementsByTagName("név").item(0);
    if(name.getTextContent().equals(oldName)) {
        //A név és a születési év átírása és előtte-utána adatok kiírása
        System.out.println("ELŐTTE:"+DomReadYd11nl.formatElement(owner, 0));
        owner.insertBefore(document.createComment("Átírt név és születési év"),
            name);
        name.setTextContent(newName);
        Element birthYear = (Element)owner.getElementsByTagName("szül_év").item(0);
        birthYear.setTextContent(newBirthYear);
        System.out.println("UTÁNA:"+DomReadYd11nl.formatElement(owner, 0));
    }
}
}

//Egy adott sorszámú tréner adott sorszámú email címének törlése, ha létezik
private static void deleteTrainerEmail(Document document, int trainerNum, int emailNum) {
    //Az adott sorszámú tréner megkeresése
    Element root = document.getDocumentElement();
    NodeList trainers = root.getElementsByTagName("tréner");
    if(trainers.getLength()>= trainerNum) {
        Element trainer = (Element)trainers.item(trainerNum-1);
        //Tréner kiírása változtatás előtt
        System.out.println("ELŐTTE:"+DomReadYd11nl.formatElement(trainer, 0));
        //Adott sorszámú email megkeresése
        Element contact = (Element)trainer.getElementsByTagName("elérhetőség").item(0);
        NodeList emails = contact.getElementsByTagName("email");
        if(emails.getLength()>= emailNum) {
            Element email = (Element)emails.item(emailNum-1);
            //Email törlése
            contact.insertBefore(document.createComment("Törölt email helye"), email);
            contact.removeChild(email);
        }
        //Tréner kiírása változtatás után
        System.out.println("UTÁNA:"+DomReadYd11nl.formatElement(trainer, 0));
    }
}

//Foglalkozás felvétele adott adatokkal, automatikusan generált fkód értékkel
private static void insertTraining(Document document, String trainerId, String[] themes,
String place, String date, String begin, String end) {
    //Új foglalkozás kódjának kiszámítása
    int maxCode = 1;
    Node next = null;
    Element root = document.getDocumentElement();
    NodeList trainings = root.getElementsByTagName("foglalkozás");

```

```

for(int i=0; i<trainings.getLength(); i++) {
    Element training = (Element)trainings.item(i);
    int id = Integer.parseInt(training.getAttribute("fkód").trim());
    if(id>maxCode) {
        maxCode = id;
    }
    //A ciklus végére a nextben az utolsó foglalkozásra rákövetkező elem lesz
    next = training.getNextSibling();

}
String trainingId = Integer.toString(maxCode + 1);
//Új foglalkozás létrehozása
Element newTraining = document.createElement("foglalkozás");
newTraining.setAttribute("fkód", trainingId);
newTraining.setAttribute("vezető", trainerId);
for (String t : themes) {
    newTraining.appendChild(createTextElement(document, "téma", t.trim()));
}
newTraining.appendChild(createTextElement(document, "helyszín", place));
Element idopontElement = document.createElement("időpont");
idopontElement.appendChild(createTextElement(document, "dátum", date));
idopontElement.appendChild(createTextElement(document, "kezdet", begin));
idopontElement.appendChild(createTextElement(document, "vég", end));
newTraining.appendChild(idopontElement);
//Beillesztés a foglalkozások közé az utolsó helyre, a next elé
root.insertBefore(document.createComment("Új foglalkozás"), next);
root.insertBefore(newTraining, next);
//Az új foglalkozás adatainak kiírása
System.out.println("ÚJ FOGLALKOZÁS:"+DomReadYd11n1.formatElement(newTraining, 0));
}

//Text node készítése
private static Element createTextElement(Document document, String tagName, String
textContent) {
    Element element = document.createElement(tagName);
    element.appendChild(document.createTextNode(textContent));
    return element;
}
}

```

2c)

Adatlekérdezés

A DomQueryYd11nl osztály meghatározott lekérdezéseket hajt végre a beolvasott XML dokumentumra vonatkozóan, melyek eredményeit a konzolra írja ki.

A következő 5+1 lekérdezést valósítottam meg, egyszerűtől az összetettebb felé haladva:

1. A 2022-ben vagy utána született kutyák adatainak kiírása (*szelekció*)
2. Az aktív tagok számának kiírása (*count típusú aggregáció*)
3. A trénerek óraber-átlagának kiszámítása és kiírása (*avg típusú aggregáció*)
4. Az összes kiosztott jutalom típusának kiírása lista formátumban, ismétlődések nélkül (*„distinct” szelekció*)
5. Annak a foglalkozásnak (Azoknak a foglalkozásoknak) az adatai, amelye(ke)n a legtöbb kutya vett részt – a legmagasabb résztvevőszám feltüntetése, majd az érintett foglalkozások részletes adatainak strukturált kiírása (*részvétel – foglalkozás összekapcsolás*)
6. Az adott nevű gazdi összes óradíj tartozásának kiszámítása és kiírása, amennyiben normál tarifa = 3500 Ft/óra, kedvezményes tarifa = 2000 Ft/óra (*gazdi – kutya – tagság és kutya – részvétel – foglalkozás összekapcsolások; összetett aggregáció*)

Output a konzolon:

1. lekérdezés:

A 2022-ben vagy utána született kutyák adatai:

```
<kutya gazdi="11" kkód="1" tagság="1000">
  <név>Bizsu</név>
  <fajta>yorkshire terrier</fajta>
  <szül_dátum>2022-10-10</szül_dátum>
  <nem>kan</nem>
</kutya>
<kutya gazdi="12" kkód="3" tagság="1001">
  <név>Bella</név>
  <fajta>magyar vizsla</fajta>
  <szül_dátum>2023-04-23</szül_dátum>
  <nem>szuka</nem>
</kutya>
```

2. lekérdezés:

Az aktív tagok száma: 3

3. lekérdezés:

A trénerek átlagos órábéré: 3217 Ft/óra

4. lekérdezés:

A kiosztott jutalmak típusai: [Rágócsont, Purina száraz kutyatáp, Kislabda, Játszókötél]

5. lekérdezés:

Azoknak a foglalkozásoknak az adatai, melyeken a legtöbben vettek részt:

A legtöbb résztvevő: 3db.

```
<foglalkozás fkód="103" vezető="51">
  <téma>engedelmesség tréning</téma>
  <helyszín>akadálypálya</helyszín>
  <időpont>
    <dátum>2023-11-02</dátum>
    <kezdet>10</kezdet>
    <vég>12</vég>
```

```
</időpont>
</foglalkozás>
```

6. lekérdezés:

Kiss Katalin összes tartozása: 47500 Ft

Az osztály kódja:

```
package hu.domparsa.yd11n1;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

import javax.xml.parsers.ParserConfigurationException;

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

public class DomQueryYd11n1 {
    public static void main(String[] args) {
        try {
            Document document = DomReadYd11n1.parseXML("XMLYd11n1.xml");

            // 1. A 2022-ben vagy utána született kutyák adatai
            String dogData = getDogNamesBornAfter(document, 2022);
            System.out.println("\u001B[31m1. lekérdezés:\u001B[30m");
            System.out.println("A 2022-ben vagy utána született kutyák adatai: " + dogData);

            // 2. Az aktív tagságok száma
            System.out.println("\n\u001B[31m2. lekérdezés:\u001B[30m");
            System.out.println("Az aktív tagok száma: " + countActiveMembers(document));

            // 3. A trénerék órabéreinek átlaga
            System.out.println("\n\u001B[31m3. lekérdezés:\u001B[30m");
            System.out.println("A trénerék átlagos órabére: " + getAvgTrainerWage(document) +
                " Ft/óra");

            // 4. Az összes kiosztott jutalom típusa
            List<String> prizes = getAllPrizes(document);
            System.out.println("\n\u001B[31m4. lekérdezés:\u001B[30m");
            System.out.println("A kiosztott jutalmak típusai: "+prizes);

            // 5. Annak a foglalkozásnak (Azoknak a foglalkozásoknak) az adatai, amin a
            // legtöbb résztvevő vett részt
            System.out.println("\n\u001B[31m5. lekérdezés:\u001B[30m");
            System.out.println("Azoknak a foglalkozásoknak az adatai, melyeken a legtöbb
                résztvevő vett részt:");
            System.out.println(getMostPopularTrainings(document));
```

```

        //6. Az adott nevű gazdi összes óradíj tartozása (normál tarifa = 3500 Ft,
        kedvezményes tarifa = 2000 Ft/óra)
        System.out.println("\n\u001B[31m6. lekérdezés:\u001B[30m");
        System.out.println("Kiss Katalin összes tartozása: "+getDebt(document, "Kiss
        Katalin")+ " Ft");

    } catch (IOException | SAXException | ParserConfigurationException e) {
        System.out.println(e.getMessage());
    }
}

//Adott évet követően született kutyák adatai strukturáltan
private static String getDogNamesBornAfter(Document document, int year) {
    String output = "";
    Element root = document.getDocumentElement();
    //Ciklus az összes kutyára
    NodeList dogs = root.getElementsByTagName("kutya");
    for (int i = 0; i < dogs.getLength(); i++) {
        Element dog = (Element) dogs.item(i);
        Element birthDateEl = (Element) dog.getElementsByTagName("szül_dátum").item(0);
        //A születési év kivétele a kutya születési dátumából
        int birthYear = Integer.parseInt(
            birthDateEl.getTextContent().split("-")[0].trim());
        //Ha a születési évnél nem nagyobb a paraméterként megadott év, akkor a kutya
        adatai a kimenetbe kerülnek
        if (birthYear >= year) {
            output+=DomReadYd11nl.formatElement(dog, 0);
        }
    }
    return output;
}

//Aktív tagságú kutyák száma
private static int countActiveMembers(Document document) {
    int counter = 0;
    Element root = document.getDocumentElement();
    // Ciklus az összes tagságra
    NodeList memberships = root.getElementsByTagName("tagság");
    for (int i = 0; i < memberships.getLength(); i++) {
        Element membership = (Element) memberships.item(i);
        // Ha az aktív attribútum értéke nem nem, a számláló növelése
        // (üres is lehet, alapértelmezetten igen)
        if (!membership.getAttribute("aktív").equals("nem")) {
            counter++;
        }
    }
    return counter;
}

```



```

//A trénerek átlagfizetése Ft/órában megadva
private static int getAvgTrainerWage(Document document) {
    int trainerCount = 0;
    int wageSum = 0;
    Element root = document.getDocumentElement();
    NodeList trainers = root.getElementsByTagName("tréner");
    // Trénerek számának meghatározása
    trainerCount = trainers.getLength();
    // Ciklus az összes trénerre
    for (int i = 0; i < trainers.getLength(); i++) {
        Element trainer = (Element) trainers.item(i);
        //Tréner órabérének hozzáadása az összeghez
        Element wage = (Element) trainer.getElementsByTagName("órabér").item(0);
        wageSum += Integer.parseInt(wage.getTextContent().trim());
    }
    //Átlag számítás, felfelé kerekítés egészre
    return (int) Math.ceil(wageSum / (1.0 * trainerCount));
}

//Az összes jutalom típus kilistázása
private static List<String> getAllPrizes(Document document) {
    List<String> allPrizes = new ArrayList<>();
    Element root = document.getDocumentElement();
    NodeList participations = root.getElementsByTagName("részvétel");
    //Ciklus az összes részvételre
    for (int i = 0; i < participations.getLength(); i++) {
        Element participation = (Element) participations.item(i);
        //Ha létezik jutalom gyerekelem, és a tartalma még nincs a listában, akkor
        felvétel
        NodeList prizes = participation.getElementsByTagName("jutalom");
        if (prizes.getLength()>0) {
            Element prize = (Element) prizes.item(0);
            if(!allPrizes.contains(prize.getTextContent())) {
                allPrizes.add(prize.getTextContent());
            }
        }
    }
    return allPrizes;
}

//A legtöbb kutya részvételével lezajlott foglalkozások formázott kiírása
private static String getMostPopularTrainings(Document document) {
    String output = "";
    Element root = document.getDocumentElement();
    NodeList trainings = root.getElementsByTagName("foglalkozás");
    // Megkeresni a részvételszám maximumát
    int max = 0;
    for (int i = 0; i < trainings.getLength(); i++) {
        Element training = (Element) trainings.item(i);
    }
}

```

```

        int participationCount = getNumberOfParticipants(document,
training.getAttribute("fkód"));
        if (participationCount > max) {
            max = participationCount;
        }
    }
    output += "A legtöbb résztvevő: " + max + "db.";
    // Megkeresni azokat a foglalkozásokat, amikre a maximummal egyenlő volt a
    részvételszám
    for (int i = 0; i < trainings.getLength(); i++) {
        Element training = (Element) trainings.item(i);
        int participationCount = getNumberOfParticipants(document,
training.getAttribute("fkód"));
        if (participationCount == max) {
            //Felvenni a foglalkozást formázva a kimeneti stringbe
            output += DomReadYd11nl.formatElement(training, 0);
        }
    }
    return output;
}

//Az adott foglalkozáson résztvevő kutyák száma
private static int getNumberOfParticipants(Document document, String trainingId) {
    int counter = 0;
    Element root = document.getDocumentElement();
    NodeList participations = root.getElementsByTagName("részvétel");
    // Ciklus az összes részvételre
    for (int i = 0; i < participations.getLength(); i++) {
        Element participation = (Element) participations.item(i);
        // Ha a megadott foglalkozásra vonatkozik a részvétel, a számláló növelése
        if (participation.getAttribute("foglalkozás").equals(trainingId)) {
            counter++;
        }
    }
    return counter;
}

//Adott nevű gazdi tartozásának kiszámítása
private static Integer getDebt(Document document, String name) {
    int DISCOUNT_RATE = 2000;
    int NORMAL_RATE = 3500;
    int debt = 0;
    //Gazdi azonosító lekérése
    String ownerId = getOwnerId(document, name);
    if(ownerId == null) return null;
    //Ciklus a gazdi kutyáira
    List<String> dogIds = getOwnedDogIds(document, ownerId);
    for(String dogId: dogIds) {
        //Tartozás növelése részvételi_órák*tarifa értékkel
        int rate = (isDiscounted(document, dogId) ? DISCOUNT_RATE : NORMAL_RATE);

```

```

        debt += getTotalParticipationHours(document, dogId)*rate;
    }
    return debt;
}

//Adott nevű gazdi azonosítója
private static String getOwnerId(Document document, String name) {
    Element root = document.getDocumentElement();
    //Ciklus az összes gazdira
    NodeList owners = root.getElementsByTagName("gazdi");
    for (int i = 0; i < owners.getLength(); i++) {
        Element owner = (Element)owners.item(i);
        //Ha a név a paraméterben megadott, akkor gkód visszaadása
        Element nameElement = (Element)owner.getElementsByTagName("név").item(0);
        if(name.equals(nameElement.getTextContent())) {
            return owner.getAttribute("gkód");
        }
    }
    return null;
}

//Adott azonosítójú gazdi kutyáinak azonosítói
private static List<String> getOwnedDogIds(Document document, String ownerId) {
    List<String> dogIds = new ArrayList<>();
    Element root = document.getDocumentElement();
    //Ciklus az összes kutyára
    NodeList dogs = root.getElementsByTagName("kutya");
    for (int i = 0; i < dogs.getLength(); i++) {
        Element dog = (Element)dogs.item(i);
        //Ha a gazdi kódja a paraméterben megadott, akkor kkód felvétele a kimeneti
        //listába
        if(dog.getAttribute("gazdi").equals(ownerId)) {
            dogIds.add(dog.getAttribute("kkód"));
        }
    }
    return dogIds;
}

//Egy adott azonosítójú kutyára kedvezményes tarifa vonatkozik-e
private static Boolean isDiscounted(Document document, String dogId) {
    String membershipId = null;
    Element root = document.getDocumentElement();
    //Adott kutyához tartozó tagság id megkeresése
    NodeList dogs = root.getElementsByTagName("kutya");
    for (int i = 0; i < dogs.getLength(); i++) {
        Element dog = (Element)dogs.item(i);
        if(dog.getAttribute("kkód").equals(dogId)) {
            membershipId = dog.getAttribute("tagság");
            break;
        }
    }
}

```

```

    }
    //Ciklus az összes tagságra
    NodeList memberships = root.getElementsByTagName("tagság");
    for (int i = 0; i < memberships.getLength(); i++) {
        Element membership = (Element)memberships.item(i);
        if(membership.getAttribute("tagkód").equals(membershipId)) {
            //A tarifa gyerekelem tartalmától függően visszatérés igazgal vagy hamissal
            String discount =
                membership.getElementsByTagName("tarifa").item(0).getTextContent();
            if(discount.equals("normál")) return false;
            else return true;
        }
    }
    return null;
}

//Egy adott kutya hány órányi foglalkozáson vett részt eddig összesen
private static int getTotalParticipationHours(Document document, String dogId) {
    int hours = 0;
    List<String> trainingIds = new ArrayList<>();
    Element root = document.getDocumentElement();
    //Ciklus az összes részvételre, érintett foglalkozás kódok listázása
    NodeList participations = root.getElementsByTagName("részvétel");
    for (int i = 0; i < participations.getLength(); i++) {
        Element participation = (Element) participations.item(i);
        if (participation.getAttribute("kutya").equals(dogId)) {
            trainingIds.add(participation.getAttribute("foglalkozás"));
        }
    }
    //Ciklus az összes foglalkozásra
    NodeList trainings = root.getElementsByTagName("foglalkozás");
    for (int i = 0; i < trainings.getLength(); i++) {
        Element training = (Element) trainings.item(i);
        if (trainingIds.contains(training.getAttribute("fkód"))) {
            Element time = (Element) training.getElementsByTagName("időpont").item(0);
            Element from = (Element) time.getElementsByTagName("kezdet").item(0);
            Element to = (Element) time.getElementsByTagName("vég").item(0);
            //A foglalkozás hosszának hozzáadása a teljes óraszámhoz, ha a kutya részt
            vett
            hours += Integer.parseInt(to.getTextContent())-
                Integer.parseInt(from.getTextContent());
        }
    }
    return hours;
}
}

```

2d)

Adatírás

A DomWriteYd11nl osztály createDogschool() metódusa felépíti a kutyaiskolához tartozó DOM fát, melyet aztán a DomReadYd11nl osztályt felhasználva kiír a konzolra és XML fájlba is, melynek neve *XMLYd11nl1.xml*. A különböző típusú elemek (kutya, tagság, gazdi, foglalkozás, részvétel, tréner, elérhetőség) létrehozására paraméterezhető metódusokat írtam.

Az osztály kódja:

```
package hu.domparse.yd11nl;

import java.io.File;
import java.text.ParseException;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.transform.TransformerException;
import javax.xml.transform.stream.StreamResult;

import org.w3c.dom.DOMException;
import org.w3c.dom.Document;
import org.w3c.dom.Element;

public class DomWriteYd11nl {
    public static void main(String[] args) {
        try {
            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
            DocumentBuilder builder = factory.newDocumentBuilder();
            Document document = builder.newDocument();

            // Kutyaiskola DOM fa felépítése
            createDogschool(document);

            // Kiírás konzolra és fájlba a DomRead osztályt felhasználva
            File newXmlFile = new File("XMLYd11nl1.xml");
            StreamResult xmlToWrite = new StreamResult(newXmlFile);
            DomReadYd11nl.writeDocument(document, xmlToWrite);
            StreamResult console = new StreamResult(System.out);
            System.out.println("A felépített dokumentum:\n");
            DomReadYd11nl.writeDocument(document, console);

        } catch (ParserConfigurationException | TransformerException | DOMException |
            ParseException e) {
            e.printStackTrace();
        }
    }
}
```

```

private static void createDogschool(Document document) throws DOMException,
ParseException {
    // Gyökérelem felvétele
    Element root = document.createElement("kutyaiskola");
    document.appendChild(root);

    // Kutyák felvétele
    root.appendChild(document.createComment("Kutyák, akik be vannak regisztrálva"));
    root.appendChild(createDog(document, "1", "11", "1000", "Bizsu", "yorkshire terrier",
    "2022-10-10", "kan"));
    root.appendChild(createDog(document, "2", "11", "1002", "Buxsi", "németjuhász",
    "2021-05-01", "kan"));
    root.appendChild(createDog(document, "3", "12", "1001", "Bella", "magyar vizsla",
    "2023-04-23", "szuka"));
    root.appendChild(createDog(document, "4", "13", "1003", "Mister", "afgán agár",
    "2016-09-15", "kan"));

    // Tagság felvétele
    root.appendChild(document.createComment("Kutyák tagság adatai"));
    root.appendChild(createMembership(document, "1000", "igen", "2023-03-01",
    "kedvezményes"));
    root.appendChild(createMembership(document, "1001", null, "2023-10-01", "normál"));
    root.appendChild(createMembership(document, "1002", null, "2022-02-28", "normál"));
    root.appendChild(createMembership(document, "1003", "nem", "2022-01-01",
    "kedvezményes"));

    //Gazdik felvétele
    root.appendChild(document.createComment("Gazdik adatai"));
    root.appendChild(createOwner(document, "11", "Kiss Katalin", "+36701234987", new
    String[]{"kisskat@gmail.com", "kati@kkconsulting.hu", "zoltan.kiss@gmail.com"},
    "1980"));
    root.appendChild(createOwner(document, "12", "Kovács Béláné Folt Irén",
    "+36208877665", new String[]{"incike@citromail.hu"}, "1961"));
    root.appendChild(createOwner(document, "13", "Petőfi Mór", "+36506060768", new
    String[]{"borok.mortol@gmail.com", "talprabaradlay@gmail.com"}, "1972"));

    //Foglalkozások felvétele
    root.appendChild(document.createComment("Foglalkozások"));
    root.appendChild(createTraining(document, "100", "51", new String[]{"elsősegély
    oktatás", "kutyapszichológia előadás", "engedelmesség tréning"}, "gyakorlóaréna",
    "2023-10-01", "14", "18"));
    root.appendChild(createTraining(document, "101", "53", new String[]{"agility
    foglalkozás", "activity foglalkozás"}, "akadálypálya", "2023-10-08", "15", "16"));
    root.appendChild(createTraining(document, "102", "51", new String[]{"engedelmesség
    tréning"}, "akadálypálya", "2023-10-15", "15", "17"));
    root.appendChild(createTraining(document, "103", "51", new String[]{"engedelmesség
    tréning"}, "akadálypálya", "2023-11-02", "10", "12"));

    //Részvételi adatok felvétele

```

```

root.appendChild(document.createComment("Kutya-foglalkozás kapcsoló elemek avagy
résztvételi adatok"));
root.appendChild(createParticipation(document, "1", "100", "4", null));
root.appendChild(createParticipation(document, "2", "100", "5", "Rágócsont"));
root.appendChild(createParticipation(document, "2", "101", "5", "Purina száraz
kutyatáp"));
root.appendChild(createParticipation(document, "3", "101", "2", "Rágócsont"));
root.appendChild(createParticipation(document, "1", "102", "5", "Kislabda"));
root.appendChild(createParticipation(document, "2", "102", "5", "Játszókötél"));
root.appendChild(createParticipation(document, "1", "103", "3", null));
root.appendChild(createParticipation(document, "2", "103", "3", null));
root.appendChild(createParticipation(document, "3", "103", "5", "Kislabda"));

//Trénerek felvétele
root.appendChild(document.createComment("Kutyatrénerek adatai"));
root.appendChild(createTrainer(document, "51", "Horváth Zebulon", new
String[]{"horvathz@ebiskola.hu"}, "06709186254", "kutyapszichológia", "3600"));
root.appendChild(createTrainer(document, "52", "Horváth Indigó", new
String[]{"horvathi@ebiskola.hu"}, "06704567821", null, "2800"));
root.appendChild(createTrainer(document, "53", "Polgár Annamária", new
String[]{"polgar.annam.2@gmail.com", "polgara@ebiskola.hu", "polgara@citromail.hu"},
"+36700670891", "agility", "3250"));

}

//Új kutya elem készítése
private static Element createDog(Document document, String dogId, String ownerId, String
membership, String name, String breed, String birthDate, String gender) throws
ParseException {
    Element dogElement = document.createElement("kutya");
    dogElement.setAttribute("kkód", dogId);
    dogElement.setAttribute("gazdi", ownerId);
    dogElement.setAttribute("tagság", membership);

    dogElement.appendChild(createTextElement(document, "név", name));
    dogElement.appendChild(createTextElement(document, "fajta", breed));
    dogElement.appendChild(createTextElement(document, "szül_dátum", birthDate));
    dogElement.appendChild(createTextElement(document, "nem", gender));

    return dogElement;
}

//Új tagság elem készítése
private static Element createMembership(Document document, String memberId, String
isActive, String firstDay, String rate) throws ParseException {
    Element membershipElement = document.createElement("tagság");
    membershipElement.setAttribute("tagkód", memberId);
    if (isActive != null) {
        membershipElement.setAttribute("aktív", isActive);
    }
}

```

```

        membershipElement.appendChild(createTextElement(document, "kezdőnap", firstDay));
        membershipElement.appendChild(createTextElement(document, "tarifa", rate));

        return membershipElement;
    }

    //Új gazdi elem készítése
    private static Element createOwner(Document document, String ownerId, String name, String
    phone, String[] emails, String birthYear) {
        Element ownerElement = document.createElement("gazdi");
        ownerElement.setAttribute("gkód", ownerId);

        ownerElement.appendChild(createTextElement(document, "név", name));
        ownerElement.appendChild(createContact(document, phone, emails));
        ownerElement.appendChild(createTextElement(document, "szül_év", birthYear));

        return ownerElement;
    }

    //Új foglalkozás elem készítése
    private static Element createTraining(Document document, String trainingId, String
    trainerId, String[] themes, String place, String date, String begin, String end) throws
    ParseException {
        Element trainingElement = document.createElement("foglalkozás");
        trainingElement.setAttribute("fkód", trainingId);
        trainingElement.setAttribute("vezető", trainerId);

        for (String t : themes) {
            trainingElement.appendChild(createTextElement(document, "téma", t.trim()));
        }
        trainingElement.appendChild(createTextElement(document, "helyszín", place));
        Element idopontElement = document.createElement("időpont");
        idopontElement.appendChild(createTextElement(document, "dátum", date));
        idopontElement.appendChild(createTextElement(document, "kezdet", begin));
        idopontElement.appendChild(createTextElement(document, "vég", end));
        trainingElement.appendChild(idopontElement);

        return trainingElement;
    }

    //Új részvétel elem készítése
    private static Element createParticipation(Document document, String dogId, String
    trainingId, String evaluation, String prize) {
        Element participationElement = document.createElement("részvétel");
        participationElement.setAttribute("kutya", dogId);
        participationElement.setAttribute("foglalkozás", trainingId);

        participationElement.appendChild(createTextElement(document, "értékelés",
        evaluation));
    }

```



```

        if (prize != null) {
            participationElement.appendChild(createTextElement(document, "jutalom", prize));
        }

        return participationElement;
    }

    //Új tréner elem készítése
    private static Element createTrainer(Document document, String trainerId, String name,
    String emails[], String telefon, String expertise, String wage) {
        Element trainerElement = document.createElement("tréner");
        trainerElement.setAttribute("tkód", trainerId);

        trainerElement.appendChild(createTextElement(document, "név", name));
        trainerElement.appendChild(createContact(document, telefon, emails));
        if(expertise != null) {
            trainerElement.appendChild(createTextElement(document, "szakterület",
            expertise));
        }
        trainerElement.appendChild(createTextElement(document, "órabér", wage));

        return trainerElement;
    }

    //Új elérhetőség elem készítése
    private static Element createContact(Document document, String phone, String[] emails) {
        Element contactElement = document.createElement("elérhetőség");

        for (String email : emails) {
            contactElement.appendChild(createTextElement(document, "email", email));
        }
        contactElement.appendChild(createTextElement(document, "telefon", phone));

        return contactElement;
    }

    //Új szöveges elem készítése
    private static Element createTextElement(Document document, String tagName, String
    textContent) {
        Element element = document.createElement(tagName);
        element.appendChild(document.createTextNode(textContent));
        return element;
    }
}

```