

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Hotelek XML-alapú adatbázisa

Készítette: **Kocsis Katalin**

Neptunkód: **WGOWUG**

Dátum: **2022.11.23.**

Tartalomjegyzék

A feladat leírása:	3
1. feladat	3
1a) Az adatbázis ER modellje	3
1b) adatbázis konvertálása XDM modellre	5
1c) Az XDM modell alapján XML dokumentum készítése ..	6
Az XML Dokumentum Kódja:	7
1d) Az XML dokumentum alapján XMLSchema készítése	14
Az XMLSchema Kódja:	15
2. feladat	17
2a) adatolvasás	18
Az adatolvasás Kódja: (DOMReadWGOWUG.java)	19
2b) adatlekérdezés	25
Az adatlekérdezés Kódja: (DOMQueryWGOWUG.java) ...	26
2c) adatmódosítás	29
Az adatmódosítás Kódja: (DOMModifyWGOWUG.java) ..	29

A feladat leírása:

A féléves feladat beadandó témája egy olyan XML-alapú adatbázis, amely több hotelt kezel. A hoteleknek vannak különböző beszállítói, és az adatbázisban létrehoztam egy beszállítás egyedet is, ami tartalmazza a beszállítótól származó dolgokat, és hogy mikor szállították be az adott hotelba. Minden hotelnak van egy Fogyasztásmérő egyede, és legalább egy alkalmazottja. Ebben az adatbázisban elkészült egy Foglалás egyed is, ami tartalmazza, hogy melyik hotelban milyen foglalások történtek. Végül ehhez van egy Foglалó egyed is, amely az előbb említett foglalás egyedhez kapcsolódva megmutatja a foglalók adatait is. Átfogó szemléletet, ismertetést ad az adatbázisunk, hogy különféle hoteleknek milyen kapcsolatai lesznek a fent említett egyedekkel. Ennek az adatbázisnak a megtervezéséhez készítettem egy ER modellt, amelyet a draw.io nevezetű programmal készítettem el. Majd ugyan evvel a programmal készítettem hozzá XDM modellt is, amely nagyban hozzájárult ahhoz, hogy átláthassam az általam kigondolt hotel témájú adatbázist, és meg tudjam írni XML-ben. Továbbá az XML dokumentumhoz tartozik egy XMLSchema, amelynek vannak saját típusai, ref, key, keyref, és speciális elemei. Az XML dokumentum adatainak adminisztrálása alapján készült még egy DOM program is. Ez a DOM program java nyelven íródott Eclipse szoftverkeretrendszer használatával. Ez a DOM program képes az általunk megírt XML-ben elkészített adatbázisunkat feldolgozni, vagyis be tudja olvasni az adatokat, és kiírja strukturált formában a konzolra, ill. menti TXT fájlba. Valamint adatmódosításra, és adat lekérdezésre is van lehetőség.

1. feladat

1a) Az adatbázis ER modellje

Ebben a hotel témájú ER modellben szerepel 6 egyed (**Beszállító, Hotel, Alkalmazott, Fogyasztásmérő, Foglалás, Foglалó**), és ezeknek az egyedeknek van többféle kapcsolat (**1:1; 1:N; M:N**). Az egyedeknek vannak tulajdonságai (**normál, kulcs, összetett, többértékű**). A következő pár sorban ezeket fogom ismertetni, és ábrával szemléltetni.

Az ER modellben szereplő egyedek és tulajdonságok ismertetése:

- **A Hotel egyed** az egyes hotelek Nyitva tartását, Nevét, Elérhetőségét tartalmazza. A HotelID lesz a hotel egyed **elsődleges kulcsa**.
- **A Beszállító egyed** az egyes beszállító cégek Címét (**Összetett tulajdonság**), Nevét, Elérhetőségét tartalmazza. Az Adószám lesz a Beszállító egyed **elsődleges kulcsa**.
- **A Beszállítás** a Hotel és a Beszállító egyed közötti N:M kapcsolatnak a tulajdonsága, ami az egyes beszállító cégek beszállítását mutatja. A beszállítás a Beszállított dolgok,

és beszállítás Dátumát tartalmazza. A K-B (Beszállító egyed felé) és a K-H (Hotel egyed felé) lesznek a beszállítás egyed **idegen kulcsai**.

- **Az alkalmazott egyed** az egyes alkalmazottak Nevét (**Összetett tulajdonság**), Születési idejét és helyét, Lakcímét (**Összetett tulajdonság**), Beosztását (milyen munkakörben dolgozik) tartalmazza. Az Személyigazolványszám lesz az Alkalmazott egyed **elsődleges kulcsa**.

A k-szig az Alkalmazott egyed **idegen kulcsa**.

- **A Fogyasztásmérő egyed** az egyes hotelekhez tartozó fogyasztásmérő Szolgáltatóját, órájának az állását, órájának a leolvasási dátuma (Hitelesítő dátum) tartalmazza. A Gyáriszám lesz a fogyasztásmérő egyed **elsődleges kulcsa**. A k-gysz a fogyasztásmérő egyed **idegen kulcsa**.

- **A Foglалás egyed** az adott hotelben a lefoglalt szobáknak a számát (**Többsértékű tulajdonság**), a Hotelek nevét (hogyan melyik nevű hotelben történt a foglalás), Foglалó nevét (aki lefoglalta a hotelben a szobát/kat), és a foglalás dátumát tartalmazza. A FoglалásID lesz a foglalás egyed **elsődleges kulcsa**. A K-F a foglalás egyed **idegen kulcsa**.

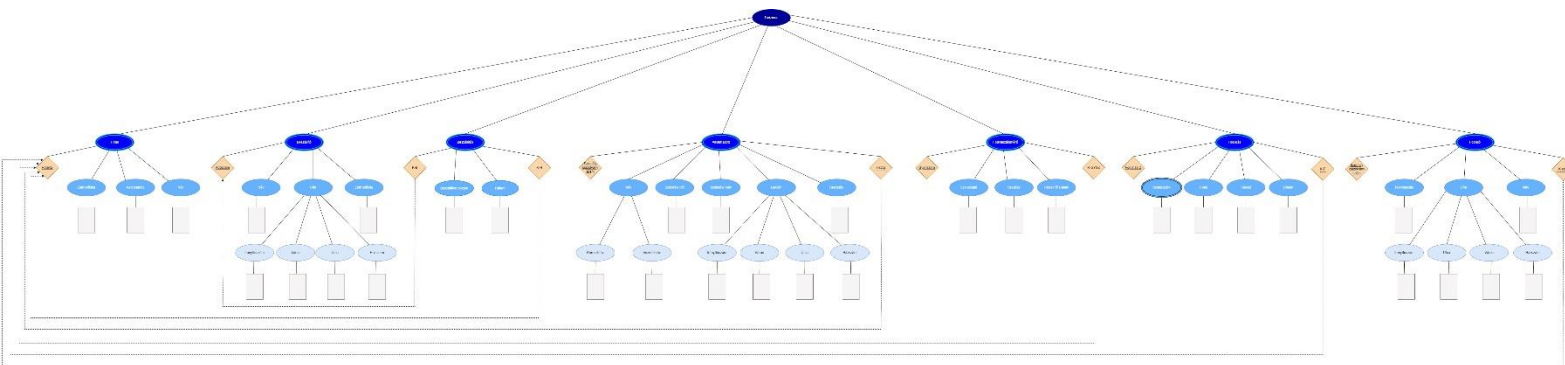
- **A Foglалó egyed** az egyes foglaláshoz tartozó személyek Telefonszámát, Címét (**Összetett tulajdonság**), Nevét tartalmazza.

A Személyigazolványszám lesz a foglaló egyed **elsődleges kulcsa**. A K-szig a foglaló egyed **idegen kulcsa**.

Az egyedek közötti kapcsolatok:

- **Beszállító reláció:** A Hotel és a Beszállító egyedek **között több a többhöz (N:M)** kapcsolat van, mivel egy hotel rendelhet több beszállítótól, valamint egy beszállító beszállíthat több hotelnek is. A kapcsolat paraméterei: a szobák fenttartásához szükséges tisztítószer, a beszállító által beszállított dolgokat jelenti, valamint a Dátum, azaz a beszállítás dátuma.
- **Alkalmazott reláció:** A Hotel egyed és az Alkalmazott egyed **egy a többhöz (1:N)** kapcsolat van, mivel egy hotel alkalmazhat több dolgozót, de egy dolgozó csak egy hotelnél dolgozik.
- **Fogyasztásmérő reláció:** A Hotel egyed és a Fogyasztásmérő egyed **egy az egyhez (1:1)** kapcsolat van, mivel egy hotelnek csak egy fogyasztásmérője lehet, és egy fogyasztásmérő is csak egy hotelhez tartozhat.
- **Foglалás reláció:** A Hotel és a Foglалás egyedek között **egy a többhöz (1:N)** kapcsolat van, mivel egy Hotelnek lehet több Foglалása, de egy foglalás csak egy hotelhez tartozhat.
- **Foglалó reláció:** A Foglалás és a Foglалó egyedek között **egy a többhöz kapcsolat van (1:N)**, mivel egy foglaló különböző hotelekbe adhatja le a foglalásait, de a foglaláshoz csak egy foglaló tartozhat.

Az adatbázis XDM-modellje:



1c) Az XDM modell alapján XML dokumentum készítése

Az XDM modell nagyban hozzájárult, és segített kialakítani az XML Dokumentumomat, és annak formázását, struktúráját. Az XML Dokumentumot megírásához Eclipse szoftverkeretrendszert használtam. Figyeltem arra, hogy **struktúráltan** legyen felépítve, megírva az XML Dokumentumom. Vagyis jól formázott, ami azt jelenti, hogy megfelel az XML specifikáció szintaktikai szabályainak. Az XML Dokumentumomnak első sora egy XML deklarációt tartalmaz, amelyet az XML specifikáció ajánl, de nem kötelező. Az XML-deklaráció általában három információt nyújt az azt tartalmazó dokumentumról: az XML-verzió információja; a használatban lévő karakterkódolás; és hogy a dokumentum önállóan áll-e, vagy külső forrásból származó információkra támaszkodik. Az XML elemeket **tag-ek** közé írtam, és minden XML elemnek rendelkezik záró címkével. A gyökér elemem neve **Turizmus**. Az XML Dokumentumban megtalálható az **XML névtér** is. Ezek módszert kínálnak az *elemnév ütközések elkerülésére*. Jelen esetben a **W3C XMLSchema névtérét** használok. A Turizmus gyökérelememnek az elemei **Beszállító, Beszállítás, Hotel, Alkalmazott, Fogasztásmérő, Foglалás, Foglалó** elemeimnek vannak **attribútumai**. Az attribútumokat a nyitótag-be írtam (*név="érték"*), ami az elsődleges kulcs és idegen kulcsokat jelöli. Az attribútum csak egyszer szerepel egy elemnél, tehát nem adtam meg azonos attribútumú elemeket. Az elemek neve *megkülönbözteti a kis-és nagybetűket, és nem tartalmaz szóközt*. A turizmus gyökérelemeinek a gyerek elemeiből **3 példányt** készítettem. Majd ezeknek az elemeknek is lesznek további gyerekelemei (pl: *Alkalmazottnál a név és a cím*) Megjegyzéseket is használtam a példányok előtt. Két tagú tartalom elem szerkezetét használtam végig a dokumentum megírása során. A két tag között beírtam a tartalmat.

Az XML Dokumentum Kódja:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<Turizmus xmlns:xsi="http://www.w3.org/2001/XMLSchema"
          xmlns="urn:Test.Namespace"
          xsi:schemaLocation="urn:Test.Namespace XMLSchemaWGOWUG.xsd">

  <!-- Gyökérelem -->

  <Hotel HotelId="1">

    <!-- Turizmus gyerekeleme, egyik egyed, hotelID tulajdonsággal (ez az elsődleges kulcs), több
    hotel(3) lehetséges turizmuson belül -->

    <Elérhetőség>

      <!-- Hotel gyerekeleme -->
      06209876543
    </Elérhetőség>

    <Nyitvatartás>

      <!-- Hotel gyerekeleme -->
      09-22
    </Nyitvatartás>

    <Név>

      <!-- Hotel gyerekeleme -->
      Király hotel
    </Név>

  </Hotel>

  <Hotel HotelId="2">

    <!-- 2.hotel -->

    <Elérhetőség> 06208765432 </Elérhetőség>

    <Nyitvatartás>00-24</Nyitvatartás>

    <Név> CsövezsItt </Név>

  </Hotel>

  <Hotel HotelId="3">

    <!-- 3.hotel -->

    <Elérhetőség> 06306876546 </Elérhetőség>

    <Nyitvatartás> 05-24 </Nyitvatartás>

    <Név> Pihenés szigete </Név>

  </Hotel>

  <Beszállító Adószám="3489371068">

    <!-- Több szállító(3) lehetséges, Turizmus gyerekeleme, ez az egyik egyed, adószám az
    elsődleges kulcs -->

    <Név>

      <!-- Beszállító gyerekeleme -->
      CleanTech
    </Név>

    <Cím>
```

```
<!-- Beszállító gyerekeleme -->

<Irányítószám>

  <!-- Cím gyerekeleme -->
    3509
  </Irányítószám>

<Város>

  <!-- Cím gyerekeleme -->
    Miskolc
  </Város>

<Utca>

  <!-- Cím gyerekeleme -->
    Fő utca
  </Utca>

<Házszám>

  <!-- Cím gyerekeleme -->
    13
  </Házszám>

</Cím>

<Elérhetőség>

  <!-- Beszállító gyerekeleme -->
    701673824
  </Elérhetőség>

</Beszállító>

<Beszállító Adószám="8316529113">

  <!-- 2.Beszállító -->

  <Név>    Ikea    </Név>

  <Cím>

    <Irányítószám>    1032    </Irányítószám>

    <Város>Budapest</Város>

    <Utca>    Arany János út    </Utca>

    <Házszám>    5    </Házszám>

  </Cím>

  <Elérhetőség>    305638633    </Elérhetőség>

</Beszállító>

<Beszállító Adószám="1835579234">

  <!-- 3.Beszállító -->

  <Név>    Media Mark    </Név>

  <Cím>

    <Irányítószám>    3500    </Irányítószám>

    <Város>Debrecen</Város>

    <Utca>    Tulipán utca    </Utca>

    <Házszám>    64    </Házszám>
```



```

</Cím>

<Elérhetőség>    207462758    </Elérhetőség>

</Beszállító>

<Beszállítás K-Bref="3489371068" K-Href="1">

    <!-- Turizmus Gyerekeleme, több is lehet belőle(3), ez az egyik egyed, K-Bref az idegen kulcs
a beszállítónak az adószáma, K-Href idegen kulcs megmondja melyik hotelbe szállít be -->

    <!-- Beszállítás gyerekeleme -->

    <Beszállított_dolgok>    Szappan    </Beszállított_dolgok>

    <Dátum>

        <!-- Beszállítás gyerekeleme -->
        11-03-2018
    </Dátum>

</Beszállítás>

<Beszállítás K-Bref="8316529113" K-Href="2">

    <!-- 2.Beszállítási, 2. hotelbe, Ikea-nak az adószáma van ott -->

    <Beszállított_dolgok>    Kanapé    </Beszállított_dolgok>

    <Dátum>    11-05-2018    </Dátum>

</Beszállítás>

<Beszállítás K-Bref="1835579234" K-Href="3">

    <!-- 3.Beszállítási, 3. hotelbe, Mediamarkt-nak az adószáma van ott -->

    <Beszállított_dolgok>    TV    </Beszállított_dolgok>

    <Dátum>    10-30-2018    </Dátum>

</Beszállítás>

<Alkalmazott SzemélyigazolványSzám="112713BR" k-szigref="1">

    <!-- Turizmus gyermekeleme, szemigszám elsődleges kulcs, k-szigref idegen kulcs azaz az
alkalmazott az lid-jű(Király Hotel) hotelbe dolgozik -->

    <Név>

        <!-- Alkalmazott Gyerekeleme -->

        <Keresztnév>

            <!-- Név gyerekeleme -->
            Anna
        </Keresztnév>

        <Vezetéknév>

            <!-- Név gyerekeleme -->
            Kiss
        </Vezetéknév>

    </Név>

    <SzületésiIdő>

        <!-- Alkalmazott Gyerekeleme -->
        1996.11.23.
    </SzületésiIdő>

    <SzületésiHely>

```

```
<!-- Alkalmazott Gyerekeleme -->
  Budapest
</SzületésiHely>

<Lakcím>

  <!-- Alkalmazott Gyerekeleme -->

  <Irányítószám>

    <!-- Lakcím gyerekeleme -->
    3576
  </Irányítószám>

  <Város>

    <!-- Lakcím gyerekeleme -->
    Sziksó
  </Város>

  <Utca>

    <!-- Lakcím gyerekeleme -->
    Hómancs
  </Utca>

  <Házzszám>

    <!-- Lakcím gyerekeleme -->
    22.
  </Házzszám>

</Lakcím>

<Beosztás>Takarító</Beosztás>
</Alkalmazott>

<Alkalmazott SzemélyigazolványSzám="214627RT" k-szigref="2">

  <!-- 2. alkalmazott, a 2.ID-jű hotelbe ( Király Hotel) dolgozik -->

  <Név>

    <Keresztnév>    Béla    </Keresztnév>

    <Vezetéknév>    Nagy    </Vezetéknév>

  </Név>

  <SzületésiIdő>    1977.04.11.    </SzületésiIdő>

  <SzületésiHely>    Békéscsaba    </SzületésiHely>

  <Lakcím>

    <Irányítószám>    5113    </Irányítószám>

    <Város>    Debrecen    </Város>

    <Utca>    Nyírfa    </Utca>

    <Házzszám>    03.    </Házzszám>

  </Lakcím>

  <Beosztás>Karbantartó</Beosztás>
</Alkalmazott>

<Alkalmazott SzemélyigazolványSzám="914581SA" k-szigref="3">

  <!-- 3. alkalmazott, a 3.ID-jű hotelbe (Pihenés Szigete) dolgozik -->
```

```
<Név>

  <Keresztnév>    Virág    </Keresztnév>

  <Vezetéknév>    Berta    </Vezetéknév>

</Név>

<SzületésiIdő>    1983.03.15.    </SzületésiIdő>

<SzületésiHely>    Budapest    </SzületésiHely>

<Lakcím>

  <Írányítószám>    1192    </Írányítószám>

  <Város>    Budapest    </Város>

  <Utca>    Teknős    </Utca>

  <Házszám>    1403.    </Házszám>

</Lakcím>

<Beosztás>    Hotel Vezető    </Beosztás>

</Alkalmazott>

<FogyasztásMérő GyáriSzám="104625" K-GYSZref="1">

  <!-- Turizmus Gyermekeleme, Mivel több hotel van több lehet belőle, viszont egy hotelhoz csak
egy tartozhat, Gyáriszám az elsődleges kulcs, K-GYSZref az idegen kulcs, 1-es id-jű (Király Hotel)
hotelhez tartozik. -->
  <!-- Fogyasztásmérő gyerekeleme -->
  <Szolgáltató>MVM</Szolgáltató>

  <Óraállás>

    <!-- Fogyasztásmérő gyerekeleme -->
    000578
  </Óraállás>

  <HitelesítőDátum>

    <!-- Fogyasztásmérő gyerekeleme -->
    2021.10.12.
  </HitelesítőDátum>

</FogyasztásMérő>

<FogyasztásMérő GyáriSzám="917249" K-GYSZref="2">

  <!-- 2. fogyasztásmérő, 2-es id-jű hotelhoz (CsövezzItt) tartozik -->
  <Szolgáltató>MVM</Szolgáltató>

  <Óraállás>    001432    </Óraállás>

  <HitelesítőDátum>    2022.03.23.    </HitelesítőDátum>

</FogyasztásMérő>

<FogyasztásMérő GyáriSzám="926712" K-GYSZref="3">

  <!-- 3. fogyasztásmérő, 3-as id-jű hotelhoz (Pihenés Szigete) tartozik -->
  <Szolgáltató>MVM</Szolgáltató>

  <Óraállás>    001558    </Óraállás>

  <HitelesítőDátum>    2022.06.30.    </HitelesítőDátum>

</FogyasztásMérő>
```

```
<Foglalás FoglалásID="111" K-Fref="1">

  <!-- Turizmus gyerekeleme, több is lehet belőle, FoglалásID elsődleges kulcs, K-Fref idegen
  kulcs, az 1-es id-jű hotelben ( Király Hotel) van foglalás -->

  <SzobaSzám>

    <!-- Foglалás gyerekeleme, több is lehet belőle(3) -->
    1355
  </SzobaSzám>

  <SzobaSzám1> 1371 </SzobaSzám1>

  <SzobaSzám2> 3467 </SzobaSzám2>
  <!-- Foglалás gyerekeleme -->
  <Hotel>Király hotel</Hotel>

  <Foglалó>

    <!-- Foglалás gyerekeleme -->
    Mészáros Kitti
  </Foglалó>

  <Dátum>

    <!-- Foglалás gyerekeleme -->
    2022.10.01.
  </Dátum>

</Foglалás>

<Foglалás FoglалásID="222" K-Fref="2">

  <!-- 2.foglалás 2id-jű (CsövezzItt) hotelbe -->

  <SzobaSzám> 2245 </SzobaSzám>

  <SzobaSzám1> 5234 </SzobaSzám1>

  <SzobaSzám2> 1298 </SzobaSzám2>

  <Hotel> CsövezzItt </Hotel>

  <Foglалó> Hanyag János </Foglалó>

  <Dátum> 2014.01.01. </Dátum>

</Foglалás>

<Foglалás FoglалásID="333" K-Fref="1">

  <!-- 3.foglалás 3id-jű (Király Hotel) hotelbe -->

  <SzobaSzám> 7012 </SzobaSzám>

  <SzobaSzám1> 3677 </SzobaSzám1>

  <SzobaSzám2> 6666 </SzobaSzám2>

  <Hotel>Király hotel</Hotel>

  <Foglалó> Móka Mihály </Foglалó>

  <Dátum> 2018.08.20. </Dátum>

</Foglалás>

<Foglалó K-szigref="1" SzemélyigazolványSzám="277023NT">

  <!-- Turizmus gyerekeleme, több is lehet belőle, Szemigszám az elsődleges kulcs, k-szigref az
  idegenkulcs, 1es id-jű (Király Hotel) hotelbe foglalt szobát -->

  <Telefonszám>
```

```
<!--Foglaló gyerekeleme -->
06701238910
</Telefonszám>

<Lakcím>

<!--Foglaló gyerekeleme -->

<Irányítószám>

<!-- Lakcím gyerekeleme -->
9823
</Irányítószám>

<Város>

<!-- Lakcím gyerekeleme -->
Múcsony
</Város>

<Utca>

<!-- Lakcím gyerekeleme -->
Major
</Utca>

<Házzsám>

<!-- Lakcím gyerekeleme -->
34
</Házzsám>

</Lakcím>

<!--Foglaló gyerekeleme -->

<Név> Mészáros Kitti </Név>
</Foglaló>

<Foglaló K-szigref="2" SzemélyigazolványSzám="123456KK">

<!-- 2.foglaló adatai, 2id-jű hotelbe foglalt (CsövezzItt) szobát(kat) -->

<Telefonszám> 06201793612 </Telefonszám>

<Lakcím>

<Irányítószám> 7159 </Irányítószám>

<Város> Badacsony </Város>

<Utca> Rózsa </Utca>

<Házzsám> 11 </Házzsám>

</Lakcím>

<Név> Hanyag János </Név>
</Foglaló>

<Foglaló K-szigref="1" SzemélyigazolványSzám="791273HI">

<!-- 3.foglaló adatai, 1id-jű hotelbe foglalt (Király hotel) szobát(kat) -->

<Telefonszám> 06309163956 </Telefonszám>

<Lakcím>

<Irányítószám> 7113 </Irányítószám>

<Város> Pécs </Város>
```

```

    <Utca>    Balaton    </Utca>

    <Házzsám>    44    </Házzsám>

</Lakcím>

    <Név>    Móka Mihály    </Név>

</Foglaló>

</Turizmus><!-- Gyökérelem vége -->

```

1d) Az XML dokumentum alapján XMLSchema készítése

A XMLSchema az előbb létrehozott XML Dokumentum szerkezetét írja le. Az XML-Schema nyelvet XML-Schema Definíciónak (**XSD**) is nevezik. Mint ahogy megemlítettem az XML Dokumentum készítésénél, a "<http://www.w3.org/2001/XMLSchema>" névteret használtam. A kódban megjegyzéseket is használtam. Az XSD-nek az első sora tartalmazza azt, hogy **jól formált** legyen a dokumentum, úgy mint az XML-nél. A **elementFormDefault="qualified"** sorral megadtam, hogy, a cél névtér elemeit a névtér előtaggal kell minősíteni. Az **attributeFormDefault="qualified"** sorral megadtam, hogy a célnévtérből származó attribútumokat a névtér-előtaggal kell minősíteni.

<xs:element> tag között van a gyökér elemem (**Turizmus**) egy **name attribútummal** megadva. Ez egy szülő elem lesz a benne található gyerekek **Beszállító, Beszállítás, Hotel, Alkalmazott, Fogyaztásmérő, Foglalás, Foglaló** elemeinek. **<xs:complexType>** tag között helyezkednek el a Turizmus gyerek elemei. Ez **egy összetett típusú elem**, olyan XML-elem, amely más elemeket és/vagy attribútumokat tartalmaz. **<xs:sequence>** tag arra szolgál, hogy a gyermekelemeknek egy szekvenciába jelenjenek meg. Minden gyermekelem 0-tól tetszőleges számú alkalommal előfordulhat. Mivel mindegyik Turizmus gyerekelemnek **3 példányt adtam meg**, így ezek közé a tagok közé kell tenni. A turizmus gyerekelemeit (pl: ref="Hotel") **egy ref attribútummal láttam** el, ami a másik elem nevére utal. A ref attribútum tartalmazhat egy névtér előtagot is. A turizmus gyerekelemei megadása után a Hotel elemnek készítettem el a gyerekelemeit, ugyanúgy **<xs:complexType>** tag között, mivel más elemeket és/vagy attribútumokat tartalmaz. **<xs:attribute name="HotelId" type="xs:string"/>** lesz a Hotel egyednek az **egyedi azonosítója, ez attribútum**. Továbbá megadtam **<xs:key>** tagok között minden turizmus gyermekelemeire az **elsődleges kulcsokat** és amelyik Turizmus elem gyerekeleme használ idegenkulcsot (**Beszállítás, Alkalmazott, Fogyaztásmérő, Foglalás, Foglaló**) azokat **<xs:keyref>** tag között adtam meg. Mivel a néhány Turizmus gyerekelemnek van további gyerekelemei, azokat is ugyanúgy elkészítettem **<xs:element>** **<xs:complexType>** tagok között. (pl: Turizmus Hotel nevű gyerekelemnek van egy Név nevű Gyerekeleme és a Név nevű gyerekelemnek vannak Keresztnév, és Vezetéknév gyerekeleme). Ezen logika mentén végig haladtam a Turizmus gyerekelemein. A **<xs:schema/>** taggel zártam az XSD Dokumentumomat, ami az XML-Séma gyökérelem végét jelenti.

Az XMLSchema Kódja:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="urn:Test.Namespace"
  xmlns="urn:Test.Namespace"
  elementFormDefault="qualified"
  attributeFormDefault="qualified"> <!-- XML-Séma gyökéreleme, tartalmazhat néhány attribútumot.
sémában használt elemek és adattípusok a "http://www.w3.org/2001/XMLSchema" névtérből származnak -->
  <xs:element name="Turizmus"> <!--Az elem neve Turizmus, a name az attribútum, szülőelem a sémaelem --
  >
    <xs:complexType> <!-- összetett típusú elem, olyan XML-elem, amely más elemeket és/vagy
attribútumokat tartalmaz. -->
      <xs:sequence> <!-- a gyermekelemeknek egy szekvenciában kell megjelenniük. Minden gyermekelem 0-
től tetszőleges számú alkalommal előfordulhat. -->
        <xs:element ref="Hotel" maxOccurs="unbounded"/> <!-- Egy másik elem nevére utal. A ref
attribútum tartalmazhat egy névtér előtagot. -->
        <xs:element ref="Beszállító" maxOccurs="unbounded"/> <!-- maxOccurs :Megadja, hogy ez az elem
hányszor fordulhat elő a szülőelemben. -->
        <xs:element ref="Beszállítás" maxOccurs="unbounded"/>
        <xs:element ref="Alkalmazott" maxOccurs="unbounded"/>
        <xs:element ref="FogyasztásMérő" maxOccurs="unbounded"/>
        <xs:element ref="Foglalás" maxOccurs="unbounded"/>
        <xs:element ref="Foglaló" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Hotel"> <!-- Hotel egyed-->
    <xs:complexType mixed="true"> <!-- szöveget és egyéb elemeket is tartalmaz -->
      <xs:sequence minOccurs="0"> <!-- minOccurs: Megadja, hogy a sorozatelem hányszor fordulhat elő
minimálisan a szülőelemben -->
        <xs:element name="Elérhetőség" type="xs:string"/> <!-- hotelnek a gyerekeleme, típusa: string -
-->
        <xs:element name="Nyitvatartás" type="xs:string"/>
        <xs:element ref="Név" minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="HotelId" type="xs:string"/> <!-- Hotel egyednek az egyedi azonosítója, ez
attribútum -->
    </xs:complexType>
    <xs:key name="HKey">
      <xs:selector xpath="Hotel" />
      <xs:field xpath="@HotelId" />
    </xs:key>
  </xs:element>
  <xs:element name="Név"> <!-- Név elem-->
    <xs:complexType mixed="true">
      <xs:sequence minOccurs="0"> <!-- sorozatelem a szülőelemben, a névbe a keresztnév és vezetéknév
van -->
        <xs:element name="Keresztnév" type="xs:string"/>
        <xs:element name="Vezetéknév" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Beszállító"> <!-- Beszállító egyed-->
    <xs:complexType>
      <xs:sequence> <!-- sorozatelem a szülőelemben, a beszállító egyednek név, cím és elérhetőség
gyereke van -->
        <xs:element ref="Név"/>
        <xs:element ref="Cím"/>
        <xs:element name="Elérhetőség" type="xs:string"/>
      </xs:sequence>
      <xs:attribute name="Adószám" type="xs:string" use="required"/> <!-- beszállítónak az elsődleges
kulcs, attribútuma az adószám -->
    </xs:complexType>
    <xs:key name="BKey">
      <xs:selector xpath="Beszállító" />
      <xs:field xpath="@Adószám" />
    </xs:key>
  </xs:element>
  <xs:element name="Cím">
    <xs:complexType>
```

```

    <xs:sequence> <!-- sorozatelem a szülőelemben, a címbe az irányítószám, város, utca, házszám
gyerekelem van -->
    <xs:element name="Irányítószám" type="xs:string"/>
    <xs:element name="Város" type="xs:string"/>
    <xs:element name="Utca" type="xs:string"/>
    <xs:element name="Házszám" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Beszállítás" <!-- Beszállítás egyed-->
  <xs:complexType>
    <xs:sequence> <!-- Beszállítás gyerekelemei -->
      <xs:element name="Beszállított_dolgok" type="xs:string"/>
      <xs:element name="Dátum" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="K-Bref" type="xs:string" use="required"/> <!-- 2 idegenkulcs, mint attributum
K-Bref, és K-Href -->
    <xs:attribute name="K-Href" type="xs:string" use="required"/>
  </xs:complexType>
    <xs:keyref name="BKeyRef" refer="BKey">
      <xs:selector xpath="Beszállítás" />
      <xs:field xpath="@K-Bref" />
    </xs:keyref>
    <xs:keyref name="BHKeyRef" refer="HKey">
      <xs:selector xpath="Beszállítás" />
      <xs:field xpath="@K-Href" />
    </xs:keyref>
  </xs:element>
<xs:element name="Alkalmazott" <!-- Alkalmazott egyed-->
  <xs:complexType>
    <xs:sequence> <!-- Alkalmazott gyerekelemei -->
      <xs:element ref="Név"/>
      <xs:element name="SzületésiIdő" type="xs:string"/>
      <xs:element name="SzületésiHely" type="xs:string"/>
      <xs:element ref="Lakcím"/>
      <xs:element name="Beosztás" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="SzemélyigazolványSzám" type="xs:string" use="required"/> <!-- Alkalmazott
elsődleges kulcsa,attributuma a szem.ig. szám -->
    <xs:attribute name="k-szigref" type="xs:string" use="required"/> <!-- Alkalmazott idegen kulcsa,
attributuma k-szigref -->
  </xs:complexType>
    <xs:key name="AKey">
      <xs:selector xpath="Alkalmazott" />
      <xs:field xpath="@SzemélyigazolványSzám" />
    </xs:key>
    <xs:keyref name="AKeyRef" refer="HKey">
      <xs:selector xpath="Alkalmazott" />
      <xs:field xpath="@k-szigref" />
    </xs:keyref>
  </xs:element>
<xs:element name="Lakcím" <!-- Alkalmazott gyerekeleme a Lakcím és a Lakcím szülő elem itt, és neki
lesznek gyerekelemei-->
  <xs:complexType>
    <xs:sequence> <!-- Lakcím gyerekelemei -->
      <xs:element name="Irányítószám" type="xs:string"/>
      <xs:element name="Város" type="xs:string"/>
      <xs:element name="Utca" type="xs:string"/>
      <xs:element name="Házszám" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="FogyasztásMérő" <!-- Fogyasztásmérő egyed-->
  <xs:complexType>
    <xs:sequence> <!-- Fogyasztásmérő gyerekelemei -->
      <xs:element name="Szolgáltató" type="xs:string"/>
      <xs:element name="Óraállás" type="xs:string"/>
      <xs:element name="HitelesítőDátum" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="GyáriSzám" type="xs:string" use="required"/> <!-- Fogyasztásmérő attribútumai
-->
    <xs:attribute name="K-GYSZref" type="xs:string" use="required"/>
  </xs:complexType>
    <xs:key name="FKey">

```



```

        <xs:selector xpath="FogyasztásMérő" />
        <xs:field xpath="@GyáriSzám" />
      </xs:key>
      <xs:keyref name="FKeyRef" refer="HKey">
        <xs:selector xpath="FogyasztásMérő" />
        <xs:field xpath="@K-GYSZref" />
      </xs:keyref>
    </xs:element>
    <xs:element name="Foglalás" <!-- Foglalás egyed -->
      <xs:complexType>
        <xs:sequence> <!-- Foglalás egyed gyerekelemei -->
          <xs:element name="SzobaSzám" type="xs:string"/>
          <xs:element name="SzobaSzám1" type="xs:string"/>
          <xs:element name="SzobaSzám2" type="xs:string"/>
          <xs:element ref="Hotel"/>
          <xs:element ref="Foglaló"/>
          <xs:element name="Dátum" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="FoglalásID" type="xs:string" use="required"/> <!-- Foglalás egyed
        attributumai -->
        <xs:attribute name="K-Fref" type="xs:string" use="required"/>
      </xs:complexType>
      <xs:key name="FogKey">
        <xs:selector xpath="Foglalás" />
        <xs:field xpath="@FoglalásID" />
      </xs:key>
      <xs:keyref name="FogKeyRef" refer="HKey">
        <xs:selector xpath="Foglalás" />
        <xs:field xpath="@K-Fref" />
      </xs:keyref>
    </xs:element>
    <xs:element name="Foglaló" <!-- Foglaló egyed -->
      <xs:complexType mixed="true">
        <xs:sequence> <!-- Foglaló egyed gyerekelemei -->
          <xs:element name="Telefonszám" type="xs:string"/>
          <xs:element ref="Lakcím"/>
          <xs:element ref="Név"/>
        </xs:sequence>
        <xs:attribute name="K-szigref" type="xs:string"/> <!-- Foglaló egyed attributumai -->
        <xs:attribute name="SzemélyigazolványSzám" type="xs:string"/>
      </xs:complexType>
      <xs:key name="FoglKey">
        <xs:selector xpath="Foglaló" />
        <xs:field xpath="@SzemélyigazolványSzám" />
      </xs:key>
      <xs:keyref name="FoglKeyRef" refer="HKey">
        <xs:selector xpath="Foglaló" />
        <xs:field xpath="@K-szigref" />
      </xs:keyref>
    </xs:element>
  </xs:schema> <!-- XML-Séma gyökérelem vége -->

```

2. feladat

Ez a feladat egy DOM program készítését fogja bemutatni XML Dokumentum adatainak adminisztrálása alapján. Eclipse szoftverkeretrendszert használtam a megírásához JAVA nyelven. Létrehoztam egy Java projektet DOMParseWGOWUG néven, majd készítettem hozzá egy package-t, aminek a neve hu.domparse.WGOWUG. Továbbá 3 osztályt is létrehoztam, amelyek az előző feladatrészben megírt XML Dokumentumot fogják feldolgozni. A DOMReadWGOWUG nevű osztályom a teljes dokumentumot feldolgozza és kiírja struktúrált formában a konzolra, illetve menti TXT fájlba. A DOMQueryWGOWUG osztályom adatlekérdezéseket fog végezni az XML Dokumentum példányaiból, és azt kiírja a konzolra. A DOMModifyWGOWUG nevű osztályom adatmódosításokat végez el az XML Dokumentum példányaiból és azt ki is írja a konzolra. A kódban megjegyzéseket is használtam.

2a) adatolvasás

A DOMReadWGOWUG nevű osztály fogja az adatolvasást elvégezni. Kezdetben importálom a szükséges csomagokat és osztályait. Utána készítek egy olyan objektumot ami közvetlenül beolvassa az XML Dokumentumot. Következő lépésben elkészítem a dokumentum olvasót, ezt a DokumentumBuilder példányosításával létrejön egy dokumentum builder, majd ez már fogja tartalmazni azt a metódust aminek a segítségével elkezdem magát a dokumentumot feldolgozni. Utána ellenőrzöm, validálom az xml dokumentumot. Ha helyes a validálás akkor elkezdem kiíratatni a gyökéreletemet. A gyökérelem kiírására a javara jellemző metódussal valósul meg. (println) Ez megnézi, hogy a dokumentumon belül melyik elem tartalmazza a gyökér elemet, és kiírja a konzolra. Az egyes gyökérelememnek (Turizmus) a gyerekeinek készítettem egy listát. Kezdetben összegyűjtöttem a Hotel nevű példányoknak a node List-jét. Itt a gyerekelemeket gyűjtjük össze tulajdonképpen. Ezt egy for ciklussal valósítom meg. Végig megy a gyerekelemeknek a listáján és le is kérdezzük ezt. Majd kiíratom az aktuális gyerekelemet. Mivel a Hotel gyerekelemhez tartozik attribútum, ezért meg kell keresni. Ezt a getAttribute fogja elvégezni, ez fogja lekérni a Hotelnek a HotelID-jét, azaz az attribútumát. Ezek után jön a következő lépés. Az egyik gyerekelemet úgy hívják hogy „Elérhetőség”, a másikat „Nyitva tartás”, és a harmadik pedig „Név” lesz. Ezeket megnézzük, hogy az 1-es,2-es,3-as (mert 3 példány van a hotelekből) ID-jű hotelen belül az 1.,2.,3(Elérhetőség, Nyitva tartás, név). gyerekelemeket, és azt hívjuk le, és nem csak az adott gyerekelemnek a nevét jelöltem meg, hanem lekértem a hozzá tartozó tartalmat is. (getTextContent()). Tehát idáig megkaptuk a hotelnek a 3 gyerekelemnek a tartalmát. A kapott információk alapján ezeket fogjuk kiíratatni a konzolra. (System.out.printf). A Turizmus Hotel gyerekelemét sikerült konzolra kiírni, és a turizmus többi gyerekelemén is végig megyünk ugyanezen logika mentén. Volt arról szó is, hogy nem csak konzolra írja ki az xml-t hanem TXT fájlba is. Erre külön létrehoztam egy „KiírásTXTFájlba” nevű metódust a main metódus mellé. Ebben a metódusban egy transzformátort fogok használni. (*TransformerFactory transformerFactory = TransformerFactory.newInstance()*) A Transformer a forrást eredménnyé alakítja. Esetünkben a forrás az XML dokumentum, az eredmény pedig a kimeneti fájl. Következő lépésben határozzuk meg az átalakítás forrását. Itt a dokumentumunkkal fogunk létrehozni egy DOM-forrást. (*DOMSource source = new DOMSource(document);*) Ezután megadtam, hogy a transzformátor hova írja ki az átalakítás eredményét. Végül megmondjuk a transzformátornak, hogy működjön a forrásobjektumon, és adja ki a kimenetet az eredmény objektumnak. Ez létrehoz egy TXT fájlt (XMLwgowug.txt) az XML Dokumentum tartalmával.

Az adatolvasás Kódja: (DOMReadWGOWUG.java)

```
1 package hu.domparsing.WGOWUG;
2
3 import java.io.File;
4
5 import java.io.IOException;
6
7 import javax.xml.parsers.DocumentBuilder;
8 import javax.xml.parsers.DocumentBuilderFactory;
9 import javax.xml.parsers.ParserConfigurationException;
10 import javax.xml.transform.OutputKeys;
11 import javax.xml.transform.Transformer;
12 import javax.xml.transform.TransformerConfigurationException;
13 import javax.xml.transform.TransformerException;
14 import javax.xml.transform.TransformerFactory;
15 import javax.xml.transform.TransformerFactoryConfigurationError;
16 import javax.xml.transform.dom.DOMSource;
17 import javax.xml.transform.stream.StreamResult;
18
19
20 import org.w3c.dom.Document;
21 import org.w3c.dom.Element;
22 import org.w3c.dom.Node;
23 import org.w3c.dom.NodeList;
24
25 import org.xml.sax.SAXException;
26
27
28
29
30 public class DomReadWGOWUG {
31
32     public static void main(String[] args ) throws SAXException, IOException, ParserConfigurationException,
33     TransformerConfigurationException, TransformerFactoryConfigurationError, TransformerException {
34
35         File xmlDoc = new File("XMLwgowug.xml");
36
37         //példányosítjuk a DocumentBuilderFactory osztályt a statikus newInstance() metódussal
38         DocumentBuilderFactory dbFact = DocumentBuilderFactory.newInstance();
39         DocumentBuilder dBuild = dbFact.newDocumentBuilder();
40         dbFact.setValidating(true); //Validáljuk az xml-t
41
42
43         // a DocumentBuilderFactory-ből megkapjuk a DocumentBuilder-t.
44
45         Document doc = dBuild.parse(xmlDoc);
46
47         doc.getDocumentElement().normalize();
48
49         //Gyökér elem kiírása
50         System.out.println("Gyökér elem neve: " + doc.getDocumentElement().getNodeName());
51
52         // a fa megadott névvel rendelkező csomópontjainak összegyűjtése
53         // csomópontok halmaza, adattagja és egy metódusa van
54
55         //read hotel element
56         NodeList nList = doc.getElementsByTagName("Hotel");
57         // A getElementsByTagName() metódus segítségével megkapjuk a hotel elem NodeList-jét
58         // a listán for ciklussal megyünk végig
59         for(int i=0; i<nList.getLength()-1;i++)
60         {
```

```

60 {
61
62     //Lekérjük a lista aktuális elemét
63     Node nNode = nList.item(i);
64
65     System.out.println("\nEgyed neve: " + nNode.getNodeName());
66
67     if(nNode.getNodeType() == Node.ELEMENT_NODE)
68     {
69         //elementté konvertáljuk a az aktuális elemet
70         Element eElement = (Element) nNode;
71         //Lekérjük az aktuális elem attribútumjának tartalmát
72         String hid = eElement.getAttribute("HotelId");
73
74         //Lekérjük az aktuális elem gverekelemeinek tartalmát
75         Node node1 = eElement.getElementsByTagName("Elérhetőség").item(0);
76         String elerhetoseg = node1.getTextContent();
77
78         Node node2 = eElement.getElementsByTagName("Nyitvatartás").item(0);
79         String nyitvatartas = node2.getTextContent();
80
81         Node node3 = eElement.getElementsByTagName("Név").item(0);
82         String hotelneve = node3.getTextContent();
83
84         System.out.printf("Hotel ID:  %s\n" , hid);
85         System.out.printf("Hotel Elérhetősége:  %s\n" , elerhetoseg);
86         System.out.printf("Hotel Nyitvatartása:  %s\n" , nyitvatartas);
87         System.out.printf("Hotel Neve:  %s\n" , hotelneve);
88
89     }
90
91 }
92
93 }
94
95
96 nList = doc.getElementsByTagName("Beszállító");
97 // A getElementsByTagName() metódus segítségével megkapjuk a Beszállító elem NodeList-jét
98 // a listán for ciklussal megvünk végig
99 for(int i=0; i<nList.getLength();i++)
100 {
101
102     //Lekérjük a lista aktuális elemét
103     Node nNode = nList.item(i);
104
105     System.out.println("\nEgyed neve: " + nNode.getNodeName());
106
107     if(nNode.getNodeType() == Node.ELEMENT_NODE)
108     {
109         //elementté konvertáljuk a az aktuális elemet
110         Element eElement = (Element) nNode;
111         //Lekérjük az aktuális elem attribútumjának tartalmát
112         String adszam = eElement.getAttribute("Adószám");
113
114
115         //Lekérjük az aktuális elem gverekelemeinek tartalmát
116         Node node1 = eElement.getElementsByTagName("Név").item(0);

```



```

117     String besznev = node1.getTextContent();
118
119     Node node2 = eElement.getElementsByTagName("Irányítószám").item(0);
120     String borsz = node2.getTextContent();
121
122     Node node3 = eElement.getElementsByTagName("Város").item(0);
123     String bvaros = node3.getTextContent();
124
125     Node node4 = eElement.getElementsByTagName("Utca").item(0);
126     String butca = node4.getTextContent();
127
128     Node node5 = eElement.getElementsByTagName("Házszám").item(0);
129     String bhsz = node5.getTextContent();
130
131     Node node6 = eElement.getElementsByTagName("Elérhetőség").item(0);
132     String beler = node6.getTextContent();
133
134     //a cím gyerekelemeinek összefűzése egy stringbe
135     String bcim = borsz + bvaros + butca + bhsz ;
136
137     System.out.printf("Beszállító adószáma: %s\n" , adszam);
138     System.out.printf("Beszállító neve: %s\n" , besznev);
139     System.out.printf("Beszállító címe: %s\n" , bcim);
140     System.out.printf("Beszállító elérhetősége: %s\n" , beler);
141 }
142
143 }
144
145 nList = doc.getElementsByTagName("Beszállítás");
146 // A getElementsByTagName() metódus segítségével megkapjuk a Beszállítás elem NodeList-jét
147 // a listán for ciklussal megyünk végig
148 for(int i=0; i<nList.getLength();i++)
149 {
150     //lekérjük a lista aktuális elemét
151     Node nNode = nList.item(i);
152
153     System.out.println("\nEgyed neve: " + nNode.getNodeName());
154
155     if(nNode.getNodeType() == Node.ELEMENT_NODE)
156     {
157         //elementté konvertáljuk a az aktuális elemet
158         Element eElement = (Element) nNode;
159         //lekérjük az aktuális elem attribútumjának tartalmát
160         String beszadszam = eElement.getAttribute("K-Bref");
161         String hid = eElement.getAttribute("K-Href");
162
163         //lekérjük az aktuális elem gyerekelemeinek tartalmát
164         Node node1 = eElement.getElementsByTagName("Beszállított_dolgok").item(0);
165         String beszdol = node1.getTextContent();
166
167         Node node2 = eElement.getElementsByTagName("Dátum").item(0);
168         String bdatum = node2.getTextContent();
169
170         System.out.printf("Beszállító adószáma: %s\n" , beszadszam);
171         System.out.printf("Ebbe az ID-jú hotelbe szállítottak be: %s\n" , hid);
172         System.out.printf("Beszállított dolog: %s\n" , beszdol);
173

```

```
174         System.out.printf("Beszállítás ideje: %s%n" , bdatum);
175     }
176
177 }
178
179 nList = doc.getElementsByTagName("Alkalmazott");
180 // A getElementsByTagName() metódus segítségével megkapjuk az Alkalmazott elem NodeList-jét
181 // a listán for ciklussal megyünk végig
182 for(int i=0; i<nList.getLength();i++)
183 {
184
185     //Lekérjük a lista aktuális elemét
186     Node nNode = nList.item(i);
187
188     System.out.println("\nEgyed neve: " + nNode.getNodeName());
189
190     if(nNode.getNodeType() == Node.ELEMENT_NODE)
191     {
192         //elementté konvertáljuk a az aktuális elemet
193         Element eElement = (Element) nNode;
194         //Lekérjük az aktuális elem attribútumjának tartalmát
195         String alkszem = eElement.getAttribute("SzemélyigazolványSzám");
196         String hid = eElement.getAttribute("k-szigref");
197
198
199         //Lekérjük az aktuális elem gverekelemeinek tartalmát
200         Node node1 = eElement.getElementsByTagName("Keresztnév").item(0);
201         String knev = node1.getTextContent();
202
203         Node node2 = eElement.getElementsByTagName("Vezetéknév").item(0);
204         String vnev = node2.getTextContent();
205
206         //név összerakása
207         String nev = vnev + knev ;
208
209         Node node3 = eElement.getElementsByTagName("SzületésiIdő").item(0);
210         String szido = node3.getTextContent();
211
212         Node node4 = eElement.getElementsByTagName("SzületésiHely").item(0);
213         String szhely = node4.getTextContent();
214
215         Node node5 = eElement.getElementsByTagName("Irányítószám").item(0);
216         String aorsz = node5.getTextContent();
217
218         Node node6 = eElement.getElementsByTagName("Város").item(0);
219         String avaros = node6.getTextContent();
220
221         Node node7 = eElement.getElementsByTagName("Utca").item(0);
222         String autca = node7.getTextContent();
223
224         Node node8 = eElement.getElementsByTagName("Házszám").item(0);
225         String ahsz = node8.getTextContent();
226
227         //a cím gverekelemeinek összerakása egy stringbe
228         String alcim = aorsz + avaros + autca + ahsz ;
229
230         Node node9 = eElement.getElementsByTagName("Beosztás").item(0);
231         ...
```



```

231 String beosz = node9.getTextContent();
232
233
234 System.out.printf("Alkalmazott személyigazolvány száma: %s\n", alkszem);
235 System.out.printf("Ebbe az ID-jű hotelbe dolgozik az alkalmazott: %s\n", hid);
236 System.out.printf("Alkalmazott teljes neve: %s\n", nev);
237 System.out.printf("Alkalmazott születési ideje: %s\n", szido);
238 System.out.printf("Alkalmazott születési helye: %s\n", szhely);
239 System.out.printf("Alkalmazott lakcíme: %s\n", alcim);
240 System.out.printf("Alkalmazott beosztása: %s\n", beosz);
241 }
242
243 }
244
245 nList = doc.getElementsByTagName("FogyasztásMérő");
246 // A getElementsByTagName() metódus segítségével megkapjuk az FogyasztásMérő elem NodeList-jét
247 // a listán for ciklussal megvünk végig
248 for(int i=0; i<nList.getLength();i++)
249 {
250
251     //Lekérjük a lista aktuális elemét
252     Node nNode = nList.item(i);
253
254     System.out.println("\nEgyed neve: " + nNode.getNodeName());
255
256     if(nNode.getNodeType() == Node.ELEMENT_NODE)
257     {
258         //elementté konvertáljuk a az aktuális elemet
259         Element eElement = (Element) nNode;
260         //Lekérjük az aktuális elem attribútumjának tartalmát
261         String gyszam = eElement.getAttribute("GyáriSzám");
262         String hid = eElement.getAttribute("K-GYSZref");
263
264
265         //Lekérjük az aktuális elem gverekelemeinek tartalmát
266         Node node1 = eElement.getElementsByTagName("Szolgáltató").item(0);
267         String szolg = node1.getTextContent();
268
269         Node node2 = eElement.getElementsByTagName("Óraállítás").item(0);
270         String oraall = node2.getTextContent();
271
272         Node node3 = eElement.getElementsByTagName("HitelesítőDátum").item(0);
273         String hitdat = node3.getTextContent();
274
275
276         System.out.printf("Fogyasztás mérő gyári száma: %s\n", gyszam);
277         System.out.printf("Ebbe az ID-jű hotelbe van a fogyasztásmérő: %s\n", hid);
278         System.out.printf("A fogyasztásmérőt biztosító szolgáltató: %s\n", szolg);
279         System.out.printf("Óraállítás: %s\n", oraall);
280         System.out.printf("Leolvasás időpontja: %s\n", hitdat);
281
282     }
283
284 }
285
286 nList = doc.getElementsByTagName("Foglalás");
287 // A getElementsByTagName() metódus segítségével megkapjuk az Foglalás elem NodeList-jét

```

```

288 // a listán for ciklussal megyünk végig
289 for(int i=0; i<nList.getLength();i++)
290 {
291
292     //Lekérjük a lista aktuális elemét
293     Node nNode = nList.item(i);
294
295     System.out.println("\nEgyed neve: " + nNode.getNodeName());
296
297     if(nNode.getNodeType() == Node.ELEMENT_NODE)
298     {
299         //elementté konvertáljuk a az aktuális elemet
300         Element eElement = (Element) nNode;
301         //Lekérjük az aktuális elem attribútumjának tartalmát
302         String fogid = eElement.getAttribute("FoglalásID");
303         String hid = eElement.getAttribute("K-Fref");
304
305
306         //Lekérjük az aktuális elem gyerekelemeinek tartalmát
307         Node node1 = eElement.getElementsByTagName("SzobaSzám").item(0);
308         String szszam = node1.getTextContent();
309
310         Node node11 = eElement.getElementsByTagName("SzobaSzám1").item(0);
311         String szszam1 = node11.getTextContent();
312
313         Node node111 = eElement.getElementsByTagName("SzobaSzám2").item(0);
314         String szszam111 = node111.getTextContent();
315
316         Node node2 = eElement.getElementsByTagName("Hotel").item(0);
317         String hot = node2.getTextContent();
318
319         Node node3 = eElement.getElementsByTagName("Foglaló").item(0);
320         String fogl = node3.getTextContent();
321
322         Node node4 = eElement.getElementsByTagName("Dátum").item(0);
323         String fdat = node4.getTextContent();
324
325
326
327         System.out.printf("Foglalás ID: %s\n" , fogid);
328         System.out.printf("Ebbe az ID-jű hotelbe foglaltak szobát: %s\n" , hid);
329         System.out.printf("Foglalt szobák száma: %s\n %s\n %s\n" , szszam, szszam1, szszam111 );
330         System.out.printf("Ebbe a nevű hotelbe foglaltak szobákat: %s\n" , hot);
331         System.out.printf("Foglaló teljes neve: %s\n" , fogl);
332         System.out.printf("Foglalás ideje: %s\n" , fdat);
333
334     }
335
336 }
337
338 nList = doc.getElementsByTagName("Foglaló");
339 // A getElementsByTagName() metódus segítségével megkapjuk az Foglaló elem NodeList-jét
340 // a listán for ciklussal megyünk végig
341 for(int i=3; i<nList.getLength();i++)
342 {
343
344     //Lekérjük a lista aktuális elemét

```



```

345     Node nNode = nList.item(i);
346
347     System.out.println("\nEgyed neve: " + nNode.getNodeName());
348
349     if(nNode.getNodeType() == Node.ELEMENT_NODE)
350     {
351         //elementt  konvert ljuk a az aktu lis elemet
352         Element eElement = (Element) nNode;
353         //Lek rj k az aktu lis elem attrib t m j nak tartalm t
354         String fogszig = eElement.getAttribute("Szem lyigazolv nyS z m");
355         String hid = eElement.getAttribute("K-szigref");
356
357         //Lek rj k az aktu lis elem gyerekelemeinek tartalm t
358         Node node1 = eElement.getElementsByTagName("Telefons z m").item(0);
359         String ftelsz = node1.getTextContent();
360
361         Node node2 = eElement.getElementsByTagName("Ir ny t s z m").item(0);
362         String firsz = node2.getTextContent();
363
364         Node node3 = eElement.getElementsByTagName("V ros").item(0);
365         String fvaros = node3.getTextContent();
366
367         Node node4 = eElement.getElementsByTagName("Utca").item(0);
368         String futca = node4.getTextContent();
369
370         Node node5 = eElement.getElementsByTagName("H zs z m").item(0);
371         String fhsz = node5.getTextContent();
372
373
374
375         //a c m gyerekelemeinek  sszef z se egy stringbe
376         String flcim = firsz + fvaros + futca + fhsh;
377
378
379         Node node6 = eElement.getElementsByTagName("N v").item(0);
380         String fnev = node6.getTextContent();
381
382         System.out.printf("Foglal  szem lyigazolv ny s z ma: %s n", fogszig);
383         System.out.printf("Ebbe az ID-j  hotelbe foglaltak szob t: %s n", hid);
384         System.out.printf("Foglal  telefons z ma %s n", ftelsz);
385         System.out.printf("Foglal  l c me: %s n", flcim);
386         System.out.printf("Foglal  teljes neve: %s n", fnev);
387
388     }
389
390 }
391
392 }
393
394 //ki rja egy txt fileba az xml doksit
395 Ki r sTXTF jlba(doc);
396
397 }
398
399 //K l n m t dus az XML doksi ki rat s hoz TXT f jlba
400 private static void Ki r sTXTF jlba(Document doc)
401     throws TransformerFactoryConfigurationException, TransformerConfigurationException, TransformerException {
402     doc.getDocumentElement().normalize();
403     TransformerFactory transformerFactory = TransformerFactory.newInstance(); //Transzform r l trehoz sa
404     Transformer transformer = transformerFactory.newTransformer();
405     DOMSource source = new DOMSource(doc); //A forr s  s az Eredm ny megad sa
406     StreamResult result = new StreamResult(new File("XMLwgowug.txt")); //megadjuk, hogy a transzform tor hova  rja ki az  talak t s eredm ny t
407     transformer.setOutputProperty(OutputKeys.INDENT, "yes"); //megfelel en beh zza a k imenetet, mivel alap rtelmez s szerint a beh z s nulla sz k z.
408     transformer.transform(source, result); //megmondjuk a transzform tornak, hogy m k dj n a forr sobjektumon,  s adja ki a k imenetet az eredm ny objektumnak
409
410 }
411 }

```

2b) adatlek rdez s

A DOMQueryWGOWUG nev  oszt ly fogja az adat lek rdez st elv gezni. A gy k r ki r s g ugyan az a logika mint a beolvas s eset ben. 5 darab lek rdez st v geztem el. Az els  lek rdez sben azokat a fogyaszt sm r knek az adatait k ri le amiknek a szol g ltat ja az MVM. Itt  sszegy jt ttem a Fogyaszt sm r  nev  p ld nyoknak a node List-j t. Majd egy for ciklus seg ts g vel bej rtam az elemeket,  s ha a Szol g ltat s elemhez  rt  s ott az MVM a szol g ltat  akkor ki ratom a konzolra azokat a Fogyaszt sm r knek a gyerekelemeit ahol az MVM a szol g ltat . A m sodik lek rdez sben is ugyan evvel a logik val dolgoztam. Ott a bes z ll t k adatait  rja ki a konzolra akik Budapestr l száll tj k az  rukat a hotelokba. A k vetkez  lek rdez sben azokat a foglal sokat fogja kilist zni a konzolra amik a Kir ly Hotel nev  hotelba t rt ntek. Majd a k vetkez  lek rdez sben megadtam, hogy k rdezze le azokat a hoteleket amik  lland an nyitva vannak. V g l azokat az alkalmazottak adatait list zza akik takar t i munkabeoszt sba vannak. A konzolra val  ki rat st ugyan avval a m dszerrel csin ltam mint amit haszn ltunk az el z  feladatr szben a beolvas sn l.

Az adatlekérdezés Kódja: (DOMQueryWGOWUG.java)

```

1 package hu.domparse.WGOWUG;
2
3 import java.io.File;
4 import java.io.IOException;
5
6 import javax.xml.parsers.DocumentBuilder;
7 import javax.xml.parsers.DocumentBuilderFactory;
8 import javax.xml.parsers.ParserConfigurationException;
9
10 import org.w3c.dom.Document;
11 import org.w3c.dom.Element;
12 import org.w3c.dom.Node;
13 import org.w3c.dom.NodeList;
14 import org.xml.sax.SAXException;
15
16 public class DomQueryWGOWUG {
17
18     public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException {
19
20         File inputFile = new File("XMLwgowug.xml");
21
22         DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
23         DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
24
25         // a DocumentBuilderFactory-ből megkapjuk a DocumentBuilder-t.
26
27         Document doc = dBuilder.parse(inputFile);
28
29         doc.getDocumentElement().normalize();
30
31         //Gyökér elem kiírása
32         System.out.println("Gyökér elem neve: " + doc.getDocumentElement().getNodeName() + "\n");
33
34         System.out.println("-----");
35
36         System.out.println("1.Lekérdezés: " + "\n" + "Azokat a fogyasztásmérőket az adatait kéri le amiknek a szolgáltatója az MVM: \n");
37         //Kérdezze le azokat a fogyasztásmérőket amiket az MVM szolgáltató
38         NodeList nList = doc.getElementsByTagName("FogyasztásMérő");
39
40         for (int temp = 0; temp < nList.getLength(); temp++) {
41             Node nNode = nList.item(temp);
42
43             if (nNode.getNodeType() == Node.ELEMENT_NODE) {
44                 Element eElement = (Element) nNode;
45
46                 Node node0;
47
48                 node0 = eElement.getElementsByTagName("Szolgáltató").item(0);
49                 String szolg = node0.getTextContent();
50                 //Ha igen, akkor kiíratom az adataikat
51                 if ("MVM".equals(szolg)) {
52                     String gyszam = eElement.getAttribute("GyáriSzám");
53                     String hid = eElement.getAttribute("K-GYSZref");
54
55                     //Lekérjük az aktuális elem gyerekeleinek tartalmát
56                     Node node1 = eElement.getElementsByTagName("Szolgáltató").item(0);
57                     String szolgi = node1.getTextContent();
58
59                     Node node2 = eElement.getElementsByTagName("Óraállítás").item(0);
60                     String oraall = node2.getTextContent();
61
62                     Node node3 = eElement.getElementsByTagName("HitelesítőDátum").item(0);
63                     String hitdat = node3.getTextContent();
64
65                     System.out.printf("Fogyasztás mérő gyári száma: %s\n", gyszam);
66                     System.out.printf("Ebbe az ID-jű hotelbe van a fogyasztásmérő: %s\n", hid);
67                     System.out.printf("A fogyasztásmérőt biztosító szolgáltató: %s\n", szolgi);
68                     System.out.printf("Óraállítás: %s\n", oraall);
69                     System.out.printf("Leolvasás időpontja: %s\n", hitdat);
70
71                 }
72
73             }
74
75         }
76
77         System.out.println("-----");
78
79         System.out.println("2.Lekérdezés: " + "\n" + "Azokat a beszállítókat adatait kéri le akik Budapestről szállítják az árukat a hotelbokba: \n");
80         //Kérdezze le azokat a Beszállítókat akik Budapestről szállítanak be
81         NodeList nList1 = doc.getElementsByTagName("Beszállító");
82
83         for (int temp = 0; temp < nList1.getLength(); temp++) {
84             Node nNode = nList1.item(temp);
85
86             if (nNode.getNodeType() == Node.ELEMENT_NODE) {
87                 Element eElement = (Element) nNode;
88
89                 Node node0;
90
91                 node0 = eElement.getElementsByTagName("Város").item(0);
92                 String bp = node0.getTextContent();
93                 //Ha igen, akkor kiíratom az adataikat
94                 if ("Budapest".equals(bp)) {
95                     String adszam = eElement.getAttribute("Adószám");
96
97                     //Lekérjük az aktuális elem gyerekeleinek tartalmát
98                     Node node1 = eElement.getElementsByTagName("Név").item(0);
99                     String besznev = node1.getTextContent();
100
101                     Node node2 = eElement.getElementsByTagName("Irányítószám").item(0);
102                     String birsz = node2.getTextContent();
103
104                     Node node3 = eElement.getElementsByTagName("Város").item(0);
105                     String bvaros = node3.getTextContent();
106
107                     Node node4 = eElement.getElementsByTagName("Utca").item(0);
108                     String butca = node4.getTextContent();
109
110                 }
111
112             }
113
114         }
115
116     }
117 }

```

```

117         Node node5 = eElement.getElementsByTagName("Hátszám").item(0);
118         String bhsz = node5.getTextContent();
119
120
121         Node node6 = eElement.getElementsByTagName("Elérhetőség").item(0);
122         String beler = node6.getTextContent();
123
124         //a cím gyerekelemeinek összefűzése egy stringbe
125         String bcim = birsz + bvaros + butca + bhsz ;
126
127         System.out.printf("Beszállító adószáma: %s\n", adszam);
128         System.out.printf("Beszállító neve: %s\n", besznev);
129         System.out.printf("Beszállító címe: %s\n", bcim);
130         System.out.printf("Beszállító elérhetősége: %s\n", beler);
131     }
132
133
134     }
135 }
136 System.out.println("-----");
137 System.out.println("3.Lekérdezés:" + "\n" + "Azokat a foglalások adatait listázza ki amik a Király Hotel nevű hotelba történtek: \n");
138 //Kérdezze le azokat a foglalásokat amik a király hotelba történtek
139 NodeList nList11 = doc.getElementsByTagName("Foglalás");
140
141
142 for (int temp = 0; temp < nList11.getLength(); temp++) {
143     Node nNode = nList11.item(temp);
144
145     if(nNode.getNodeType() == Node.ELEMENT_NODE) {
146         Element eElement = (Element) nNode;
147
148         Node node0;
149
150         node0 = eElement.getElementsByTagName("Hotel").item(0);
151         String khot = node0.getTextContent();
152         //Ha igen, akkor kiíratom az adataikat
153         if("Király hotel".equals(khot)) {
154
155
156             String fogid = eElement.getAttribute("FoglalásID");
157             String hid = eElement.getAttribute("K-Fref");
158
159
160             //Lekérjük az aktuális elem gyerekelemeinek tartalmát
161             Node node1 = eElement.getElementsByTagName("SzobaSzám").item(0);
162             String szszam = node1.getTextContent();
163
164             Node node11 = eElement.getElementsByTagName("SzobaSzám1").item(0);
165             String szszam1 = node11.getTextContent();
166
167             Node node111 = eElement.getElementsByTagName("SzobaSzám2").item(0);
168             String szszam111 = node111.getTextContent();
169
170             Node node2 = eElement.getElementsByTagName("Hotel").item(0);
171             String hot = node2.getTextContent();
172
173             Node node3 = eElement.getElementsByTagName("Foglaló").item(0);
174             String fogl = node3.getTextContent();
175
176             Node node4 = eElement.getElementsByTagName("Dátum").item(0);
177             String fdat = node4.getTextContent();
178
179
180             System.out.printf("Foglalás ID: %s\n", fogid);
181             System.out.printf("Ebbe az ID-jú hotelba foglaltak szobát: %s\n", hid);
182             System.out.printf("Foglalt szobák száma: %s\n %s\n %s\n", szszam, szszam1, szszam111 );
183             System.out.printf("Ebbe a nevű hotelba foglaltak szobákat: %s\n", hot);
184             System.out.printf("Foglaló teljes neve: %s\n", fogl);
185             System.out.printf("Foglalás ideje: %s\n", fdat);
186
187         }
188     }
189 }
190
191
192 }
193 }
194 System.out.println("-----");
195 System.out.println("4.Lekérdezés:" + "\n" + "Azokat a Hotelek adatait listázza ki amik éjjel nappal nyitva vannak: \n");
196 //Kérdezze le azokat a hoteleket amik állandóan nyitva vannak
197 NodeList nList111 = doc.getElementsByTagName("Hotel");
198
199
200 for (int temp = 0; temp < nList111.getLength()-3; temp++) {
201     Node nNode = nList111.item(temp);
202
203     if(nNode.getNodeType() == Node.ELEMENT_NODE) {
204         Element eElement = (Element) nNode;
205
206         Node node0;
207
208         node0 = eElement.getElementsByTagName("Nyitvatartás").item(0);
209         String nyt = node0.getTextContent();
210         //Ha igen, akkor kiíratom az adataikat
211         if("00-24".equals(nyt)) {
212
213             String hid = eElement.getAttribute("HotelId");
214
215             //Lekérjük az aktuális elem gyerekelemeinek tartalmát
216             Node node1 = eElement.getElementsByTagName("Elérhetőség").item(0);
217             String elerhetoseg = node1.getTextContent();
218
219             Node node2 = eElement.getElementsByTagName("Nyitvatartás").item(0);
220             String nyitvatartas = node2.getTextContent();
221
222             Node node3 = eElement.getElementsByTagName("Név").item(0);
223             String hotelneve = node3.getTextContent();
224
225             System.out.printf("Hotel ID: %s\n", hid);
226             System.out.printf("Hotel Elérhetősége: %s\n", elerhetoseg);
227             System.out.printf("Hotel Nyitvatartása: %s\n", nyitvatartas);
228             System.out.printf("Hotel Neve: %s\n", hotelneve);
229         }
230     }
231 }

```



```

231 }
232
233
234 }
235 }System.out.println("-----");
236 System.out.println("5.Lekérdezés:" + "\n" + "Azokat az alkalmazottak adatait listázza akik takarítói munkabeosztásba vannak: \n");
237 //Kérdezze le azokat az alkalmazottakat akik takarítók
238 NodeList nList1111 = doc.getElementsByTagName("Alkalmazott");
239
240
241 for (int temp =0; temp < nList1111.getLength(); temp++) {
242     Node nNode = nList1111.item(temp);
243
244
245     if(nNode.getNodeType() == Node.ELEMENT_NODE) {
246         Element eElement = (Element) nNode;
247
248         Node node0;
249
250         node0 = eElement.getElementsByTagName("Beosztás").item(0);
251         String beo = node0.getTextContent();
252         //Ha igen, akkor kiíratom az adataikat
253         if("Takarító".equals(beo)) {
254
255             String alkszem = eElement.getAttribute("SzemélyigazolványSzám");
256             String hid = eElement.getAttribute("k-szigref");
257
258             //lekérjük az aktuális elem gverekelemeinek tartalmát
259             Node node1 = eElement.getElementsByTagName("Keresztnév").item(0);
260             String knev = node1.getTextContent();
261
262             Node node2 = eElement.getElementsByTagName("Vezetéknév").item(0);
263             String vnev = node2.getTextContent();
264
265             //név összefűzése
266             String nev = vnev + knev ;
267
268             Node node3 = eElement.getElementsByTagName("SzületésiIdő").item(0);
269             String szido = node3.getTextContent();
270
271             Node node4 = eElement.getElementsByTagName("SzületésiHely").item(0);
272             String szhely = node4.getTextContent();
273
274             Node node5 = eElement.getElementsByTagName("Irányítószám").item(0);
275             String aorsz = node5.getTextContent();
276
277             Node node6 = eElement.getElementsByTagName("Város").item(0);
278             String avaros = node6.getTextContent();
279
280             Node node7 = eElement.getElementsByTagName("Utca").item(0);
281             String autca = node7.getTextContent();
282
283             Node node8 = eElement.getElementsByTagName("Házszám").item(0);
284             String ahsz = node8.getTextContent();
285
286             //a cím gverekelemeinek összefűzése egy stringbe
287             String alcim = aorsz + avaros + autca + ahsz ;
288
289             Node node9 = eElement.getElementsByTagName("Beosztás").item(0);
290             String beosz = node9.getTextContent();
291
292             System.out.printf("Alkalmazott személyigazolvány száma: %s\n", alkszem);
293             System.out.printf("Ebbe az ID-jű hotelbe dolgozik az alkalmazott: %s\n", hid);
294             System.out.printf("Alkalmazott teljes neve: %s\n", nev);
295             System.out.printf("Alkalmazott születési ideje: %s\n", szido);
296             System.out.printf("Alkalmazott születési helye: %s\n", szhely);
297             System.out.printf("Alkalmazott lakcíme: %s\n", alcim);
298             System.out.printf("Alkalmazott beosztása: %s\n", beosz);
299
300         }
301     }
302 }
303
304 }
305
306 }
307
308 }
309
310 }
311
312 }

```

2c) adatmódosítás

A DOMModifyWGOWUG nevű osztály fogja az adatmódosítást elvégezni. Kezdetben importálom a szükséges csomagokat és osztályait. A különböző módosításokat egye egy metódusban végzem el. 6 módosítást végeztem el. Van egy olyan módosításom ami a Beszállításnál a beszállított dolgokat frissíti ajándéktárggyá. Utána lesz egy olyan módosítás ami az első foglaló nevét fogja átírni. Azután törlést végzünk el. Törölődni fog a Foglalásból a Hotel neve, valamint az Alkalmazott Lakcímét törli. Adat frissítés és törlés mellett új gyerekelemet (gyerekelem neve: Cím) is hozzáadtam a Hotel elemhez, és annak megadtam a tartalmát. Valamint a Fogyasztásmérőnél is hozzáadtam egy gyártó gyerekelemet tartalommal együtt. A konzolra kiíratást két féleképpen és 2 külön metódusban végeztem el. Egyrészt a módosult XML Dokumentumot kiírtam XML Dokumentumba (Updated_XMLWgowug.xml), majd azt visszaolvasva kiírtam a konzolra (avval a módszerrel amivel a DOMReadWGOWUG.java fájlban láthattuk), de csak a módosult elemeket. A másik megoldásban a teljes módosult XML Dokumentumot kiírtam a konzolra.

Az adatmódosítás Kódja: (DOMModifyWGOWUG.java)

```

1 package hu.domparse.WGOWUG;
2
3 import java.io.File;
4
5 public class DomModifyWGOWUG {
6
7     public static void main(String[] args) throws TransformerConfigurationException,
8         TransformerFactoryConfigurationError, TransformerException {
9         String filePath = "XMLWgowug.xml";
10         File xmlFile = new File(filePath);
11         DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
12         DocumentBuilder dBuilder;
13
14         try {
15             dBuilder = dbFactory.newDocumentBuilder();
16             Document doc = dBuilder.parse(xmlFile);
17             doc.getDocumentElement().normalize();
18
19             // Módosítja a Beszállításnál a beszállított dolgokat, mindegyiket ajándéktárggyá módosítja
20             Módosit(doc);
21
22             //Első foglaló nevének átírása
23             Módosit2(doc);
24
25             // A foglalásból törli a hotel nevét
26             Töröl(doc);
27
28             // az alkalmazottak címét törli
29             Töröl2(doc);
30
31             // hozzáad egy "cím" elemet a hotelhoz
32             Hozzáad(doc);
33
34             // hozzáad egy "gyártó" elemet a fogyasztásmérőhöz
35             Hozzáad2(doc);
36
37             KiírdXMLFájlba(doc);
38
39             Kiír(doc);
40
41             Kiír2(doc);
42
43         } catch (SAXException | ParserConfigurationException | IOException e) {
44             e.printStackTrace();
45         }
46     }
47
48     private static void Kiír2(Document doc) throws TransformerException {
49         System.out.printf("\n" + "-----"
50             + "-----" + "\n" + "Teljes módosult XML Doksí kiírása:" + "\n" + "\n");
51
52         Transformer transformer = TransformerFactory.newInstance().newTransformer();
53         //Transformer létrehozása
54         transformer.setOutputProperty(OutputKeys.ENCODING, "UTF-8");
55         //UTF 8 kódolású legyen , hogy olvasható legyen
56         transformer.setOutputProperty(OutputKeys.INDENT, "yes");
57         //megfelelően behúzza a kimenetet, mivel alapértelmezés szerint a behúzás nulla szóköz.
58         transformer.setOutputProperty("http://xml.apache.org/xslt indent-amount", "2");
59         DOMSource source = new DOMSource(doc); //A Forrás és az Eredmény megadása
60         Writer stringWriter = new StringWriter();
61         StreamResult streamResult = new StreamResult(stringWriter);
62         //megadjuk, hogy a transzformátor hova írja ki az átalakítás eredményét
63         transformer.transform(source, streamResult);
64         //megmondjuk a transzformátornak, hogy működjön a forrásobjektumon, és adja ki a kimenetet az eredmény objektumnak
65         String result = stringWriter.toString();
66         System.out.println(result);
67     }
68
69     //cím Elem hozzáadása a hotelhez
70     private static void Hozzáad(Document doc) {
71         NodeList Turizmus = doc.getElementsByTagName("Hotel");
72         Element Hotel = null;
73
74         // loop for each user
75         for (int i = 0; i < Turizmus.getLength(); i++) {
76             Hotel = (Element) Turizmus.item(i);
77             Element id = doc.createElement("Cím");
78             id.appendChild(doc.createTextNode("3571 Árnót József Attila út 76."));
79             Hotel.appendChild(id);
80         }
81     }
82 }

```

```
105     }
106 }
107
108 private static void Hozzáad2(Document doc) {
109     NodeList Turizmus = doc.getElementsByTagName("FogyasztásMérő");
110     Element FogyasztásMérő = null;
111
112     // loop for each user
113     for (int i = 0; i < Turizmus.getLength(); i++) {
114         FogyasztásMérő = (Element) Turizmus.item(i);
115         Element id = doc.createElement("Gyártó");
116         id.appendChild(doc.createTextNode("Electryc Fogyasztásmérő KFT"));
117         FogyasztásMérő.appendChild(id);
118     }
119 }
120
121
122 //Egy elem törlése foglalásból a hotel-t ( hotel neve)
123 private static void Töröl(Document doc) {
124     NodeList Turizmus = doc.getElementsByTagName("Foglalás");
125     Element Foglalás = null;
126     // loop for each user
127     for (int i = 0; i < Turizmus.getLength(); i++) {
128         Foglalás = (Element) Turizmus.item(i);
129         Node cim = Foglalás.getElementsByTagName("Hotel").item(0);
130         Foglalás.removeChild(cim);
131     }
132 }
133
134 //Egy elem törlése az alkalmazott egyedből, alkalmazottnak a lakcímét törli
135 private static void Töröl2(Document doc) {
136     NodeList Turizmus = doc.getElementsByTagName("Alkalmazott");
137     Element Alkalmazott = null;
138     // loop for each user
139     for (int i = 0; i < Turizmus.getLength(); i++) {
140         Alkalmazott = (Element) Turizmus.item(i);
141         Node Lakcím = Alkalmazott.getElementsByTagName("Lakcím").item(0);
142         Alkalmazott.removeChild(Lakcím);
143     }
144 }
145
146 //Beszállítás első elemének frissítése, módosítása
147 private static void Módosít(Document doc) {
```



```

148     NodeList Turizmus = doc.getElementsByTagName("Beszállítás");
149     Element Beszállítás = null;
150     // loop for each user
151     for (int i = 0; i < Turizmus.getLength(); i++) {
152         Beszállítás = (Element) Turizmus.item(i);
153         Node Beszállított_dolgok = Beszállítás.getElementsByTagName("Beszállított_dolgok").item(0).getFirstChild();
154         Beszállított_dolgok.setNodeValue("Ajándék tárgy");
155     }
156 }
157
158 //Mészáros Kitti nevű foglалó átírása Fehérné Nagy Klárává
159 private static void Módosít2(Document doc) {
160     NodeList Turizmus = doc.getElementsByTagName("Foglалó");
161     Element Foglалó = null;
162     // loop for each user
163     for (int i = 3; i < Turizmus.getLength()-2; i++) {
164         Foglалó = (Element) Turizmus.item(i);
165         Node Név = Foglалó.getElementsByTagName("Név").item(0).getFirstChild();
166         Név.setNodeValue("Fehérné Nagy Klára");
167     }
168 }
169
170 }
171
172 private static void KiírásXMLFájlba(Document doc)
173     throws TransformerFactoryConfigurationException, TransformerConfigurationException, TransformerException {
174     doc.getDocumentElement().normalize();
175     TransformerFactory transformerFactory = TransformerFactory.newInstance();
176     //Transzformer létrehozása
177     Transformer transformer = transformerFactory.newTransformer();
178     DOMSource source = new DOMSource(doc);
179     //A forrás és az eredmény megadása
180     StreamResult result = new StreamResult(new File("Updated_XMLwgowug.xml"));
181     //megadjuk, hogy a transzformátor hova írja ki az átalakítás eredményét
182     transformer.setOutputProperty(OutputKeys.INDENT, "yes");
183     //megfelelően behúzza a kimenetet, mivel alapértelmezés szerint a behúzás nulla szóköz.
184     transformer.transform(source, result);
185     //megmondjuk a transzformátornak, hogy működjön a forrásobjektumon, és adja ki a kimenetet az eredmény objektumnak
186 }
187
188
189 private static void Kiír(Document doc) throws ParserConfigurationException, SAXException, IOException {
190
191     File xmlDoc = new File("Updated_XMLwgowug.xml");
192
193     //példányosítjuk a DocumentBuilderFactory osztályt a statikus newInstance() metódussal
194     DocumentBuilderFactory dbFact = DocumentBuilderFactory.newInstance();
195     DocumentBuilder dBuild = dbFact.newDocumentBuilder();
196     // a DocumentBuilderFactory-ből megkapjuk a DocumentBuilder-t.
197
198     Document doc1 = dBuild.parse(xmlDoc);
199
200     doc1.getDocumentElement().normalize();
201
202     System.out.println("1.Módosítás: Hozzáad egy cím elemet a hotelhoz");
203
204     NodeList nList = doc.getElementsByTagName("Hotel");
205     // A getElementsByTagName() metódus segítségével megkapjuk a hotel elem NodeList-jét
206     // a listán for ciklussal megyünk végig
207     for(int i=0; i<nList.getLength();i++)
208     {
209
210         //Lekérjük a lista aktuális elemét
211         Node nNode = nList.item(i);
212
213         System.out.println("\nEgyed neve: " + nNode.getNodeName());
214
215         if(nNode.getNodeType() == Node.ELEMENT_NODE)
216         {
217             //elementtē konvertáljuk a az aktuális elemet
218             Element eElement = (Element) nNode;
219             //Lekérjük az aktuális elem attribútumjának tartalmát
220             String hid = eElement.getAttribute("HotelId");
221
222             System.out.printf("Hotel ID: %s\n", hid);
223             System.out.println("Elérhetőség:"+eElement.getElementsByTagName("Elérhetőség").item(0).getTextContent()
224                 +"\n" + "Nyitvatartás:"+eElement.getElementsByTagName("Nyitvatartás").item(0).getTextContent()
225                 +"\n" + "Név:"+eElement.getElementsByTagName("Név").item(0).getTextContent()+"\n" + "Cím:"
226                 +eElement.getElementsByTagName("Cím").item(0).getTextContent());
227
228         }
229     }
230 }
231

```

```

232 }
233 System.out.println("-----");
234 System.out.println("2. Módosítás: Frissíti, átírja a Beszállításnál a beszállított dolgokat");
235
236 nList = doc.getElementsByTagName("Beszállítás");
237 // A getElementsByTagName() metódus segítségével megkapjuk a Beszállítás elem NodeList-jét
238 // a listán for ciklussal megyünk végig
239 for(int i=0; i<nList.getLength();i++)
240 {
241
242     //Lekérjük a lista aktuális elemét
243     Node nNode = nList.item(i);
244
245     System.out.println("\nEgyed neve: " + nNode.getNodeName());
246
247     if(nNode.getNodeType() == Node.ELEMENT_NODE)
248     {
249         //elementtől konvertáljuk a az aktuális elemet
250         Element eElement = (Element) nNode;
251         //Lekérjük az aktuális elem attribútumjának tartalmát
252         String beszadszam = eElement.getAttribute("K-Bref");
253         String hid = eElement.getAttribute("K-Href");
254
255         System.out.printf("Beszállító adószáma: %s\n", beszadszam);
256         System.out.printf("Ebbe az ID-jű hotelbe szállítottak be: %s\n", hid);
257         System.out.println("Beszállított dolgok:" + eElement.getElementsByTagName("Beszállított dolgok").item(0).getTextContent()
258             + "\n" + "Dátum:" + eElement.getElementsByTagName("Dátum").item(0).getTextContent());
259     }
260 }
261
262 System.out.println("-----");
263 System.out.println("3. Módosítás: Az alkalmazottak címét törli");
264 nList = doc.getElementsByTagName("Alkalmazott");
265 // A getElementsByTagName() metódus segítségével megkapjuk az Alkalmazott elem NodeList-jét
266 // a listán for ciklussal megyünk végig
267 for(int i=0; i<nList.getLength();i++)
268 {
269
270     //Lekérjük a lista aktuális elemét
271     Node nNode = nList.item(i);
272
273     System.out.println("\nEgyed neve: " + nNode.getNodeName());
274
275     if(nNode.getNodeType() == Node.ELEMENT_NODE)
276     {
277         //elementtől konvertáljuk a az aktuális elemet
278         Element eElement = (Element) nNode;
279         //Lekérjük az aktuális elem attribútumjának tartalmát
280         String alkszem = eElement.getAttribute("SzemélyigazolványSzáma");
281         String hid = eElement.getAttribute("k-szigref");
282
283         System.out.printf("Alkalmazott személyigazolvány száma: %s\n", alkszem);
284         System.out.printf("Ebbe az ID-jű hotelbe dolgozik az alkalmazott: %s\n", hid);
285         System.out.println("Keresztnév:" + eElement.getElementsByTagName("Keresztnév").item(0).getTextContent() + "\n"
286             + "Vezetéknév:" + eElement.getElementsByTagName("Vezetéknév").item(0).getTextContent()
287             + "\n" + "SzületésiIdő:" + eElement.getElementsByTagName("SzületésiIdő").item(0).getTextContent() + "\n"
288             + "\n" + "SzületésiHely:" + eElement.getElementsByTagName("SzületésiHely").item(0).getTextContent()
289             /* Ha kivesszük a kommenteket, akkor hibát dob, mert az az elem már nem létezik amit ki akarunk írni vagyis a cím
290             + "\n" + "Irányítószám:" + eElement.getElementsByTagName("Irányítószám").item(0).getTextContent()
291             + "\n" + "Város:" + eElement.getElementsByTagName("Város").item(0).getTextContent()
292             + "\n" + "Utca:" + eElement.getElementsByTagName("Utca").item(0).getTextContent()
293             + "\n" + "Házszám:" + eElement.getElementsByTagName("Házszám").item(0).getTextContent() */
294             + "\n" + "Beosztás:" + eElement.getElementsByTagName("Beosztás").item(0).getTextContent());
295     }
296 }
297
298 }
299 System.out.println("-----");
300 System.out.println("4. Módosítás: hozzáad egy gyártó elemet a fogyasztásmérőhöz");
301
302 nList = doc.getElementsByTagName("FogyasztásMérő");
303 // A getElementsByTagName() metódus segítségével megkapjuk az Fogyasztásmérő elem NodeList-jét
304 // a listán for ciklussal megyünk végig
305 for(int i=0; i<nList.getLength();i++)
306 {
307
308     //Lekérjük a lista aktuális elemét
309     Node nNode = nList.item(i);
310
311     System.out.println("\nEgyed neve: " + nNode.getNodeName());
312
313     if(nNode.getNodeType() == Node.ELEMENT_NODE)
314     {
315

```



```

316 //elementt  konvert ljuk a az aktu lis elemet
317 Element eElement = (Element) nNode;
318 //Lek rj k az aktu lis elem attrib t m j nak tartalm t
319 String gyszam = eElement.getAttribute("Gy risz m");
320 String hid = eElement.getAttribute("K-GYSZref");
321
322
323 System.out.printf("Fogyaszt s m r  gy ri s ma: %s n", gyszam);
324 System.out.printf("Ebbe az ID-j  hotelbe van a fogyaszt sm r : %s n", hid);
325 System.out.println("Szolg ltat : "+eElement.getElementsByTagName("Szolg ltat ").item(0).getTextContent()+
326 "\n"+ " ra ll s: "+eElement.getElementsByTagName(" ra ll s").item(0).getTextContent()+
327 "\n"+ "Hiteles t D tum: "+eElement.getElementsByTagName("Hiteles t D tum").item(0).getTextContent()+ " \n"
328 "\n"+ "Gy rt : "+eElement.getElementsByTagName("Gy rt ").item(0).getTextContent());
329
330
331 }
332
333 System.out.println("-----");
334 System.out.println("5. M dos t s: A foglal sb l t rli a hotel nev t");
335
336 nList = doc.getElementsByTagName("Foglal s");
337 // A getElementsByTagName() met dus seg ts g vel megkapjuk az Foglal s elem NodeList-j t
338 // a list n for ciklussal megy nk v g g 
339 for(int i=0; i<nList.getLength();i++)
340 {
341
342 //Lek rj k a lista aktu lis elem t
343 Node nNode = nList.item(i);
344
345 System.out.println("\nEgyed neve: " + nNode.getNodeName());
346
347 if(nNode.getNodeType() == Node.ELEMENT_NODE)
348 {
349 //elementt  konvert ljuk a az aktu lis elemet
350 Element eElement = (Element) nNode;
351 //Lek rj k az aktu lis elem attrib t m j nak tartalm t
352 String fogid = eElement.getAttribute("Foglal sID");
353 String hid = eElement.getAttribute("K-Fref");
354
355 System.out.printf("Foglal s ID: %s n", fogid);
356 System.out.printf("Ebbe az ID-j  hotelbe foglaltak szob t: %s n", hid);
357 System.out.println("SzobaS z m: "+eElement.getElementsByTagName("SzobaS z m").item(0).getTextContent()+
358 "\n"+eElement.getElementsByTagName("SzobaS z m1").item(0).getTextContent()+
359 "\n"+eElement.getElementsByTagName("SzobaS z m2").item(0).getTextContent()+ " \n"
360 //Ha ki vessz k a kommenteket, akkor hib t dob, mert az az elem m r nem l tezik amit ki akarunk  ratni vagyis a hotel neve
361 //+"Hotel: "+eElement.getElementsByTagName("Hotel").item(0).getTextContent()+
362 "\n"+ "Foglal  neve: "+eElement.getElementsByTagName("Foglal ").item(0).getTextContent()+
363 "\n"+ "Foglal s d tuma: "+eElement.getElementsByTagName("D tum").item(0).getTextContent());
364
365 }
366
367 }
368
369 System.out.println("-----");
370 System.out.println("6. M dos t s: Els  foglal  nev nek  tir sa");
371 nList = doc.getElementsByTagName("Foglal ");
372 // A getElementsByTagName() met dus seg ts g vel megkapjuk az Foglal  elem NodeList-j t
373 // a list n for ciklussal megy nk v g g 
374 for(int i=3; i<nList.getLength();i++)
375 {
376
377 //Lek rj k a lista aktu lis elem t
378 Node nNode = nList.item(i);
379
380 System.out.println("\nEgyed neve: " + nNode.getNodeName());
381
382 if(nNode.getNodeType() == Node.ELEMENT_NODE)
383 {
384 //elementt  konvert ljuk a az aktu lis elemet
385 Element eElement = (Element) nNode;
386 //Lek rj k az aktu lis elem attrib t m j nak tartalm t
387 String fogszig = eElement.getAttribute("Szem lyigazolv nyS z m");
388 String hid = eElement.getAttribute("K-szigref");
389
390 System.out.printf("Foglal  szem lyigazolv ny s ma: %s n", fogszig);
391 System.out.printf("Ebbe az ID-j  hotelbe foglaltak szob t: %s n", hid);
392 System.out.println("Megv ltozott Egyed: " + "\n"+ "Telefons z m: "+
393 eElement.getElementsByTagName("Telefons z m").item(0).getTextContent()+
394 "\n"+ " r ny t s z m: "+eElement.getElementsByTagName(" r ny t s z m").item(0).getTextContent()+
395 "\n"+ "V ros: "+eElement.getElementsByTagName("V ros").item(0).getTextContent()+
396 "\n"+ "Utca: "+eElement.getElementsByTagName("Utca").item(0).getTextContent()+
397 "\n"+ "H zs z m: "+eElement.getElementsByTagName("H zs z m").item(0).getTextContent()+
398 "\n"+ "N v: "+eElement.getElementsByTagName("N v").item(0).getTextContent());
399
400 }
401
402 }
403
404 }
405
406

```