JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Webshop

Készítette: **Hajdu Adrián**
Neptunkód: **UY5E1L**

Dátum: 2023.12.05.

Tartalomjegyzék

Tartalom

Bevezetés

A feladat leírása:

A feladat elkészítéséhez egy webshop adatbázisát reprezentáló rendszert hoztam létre. A rendszerben nyilvántartásra kerülnek a webshop által árusított termékek, a leadott megrendelések, a megrendelt termékek, az oldalon regisztrált felhasználók (ügyfelek), illetve a céggel szerződést kötött szállítók, és az ezekhez tartozó adatok.

A felhasználók az oldalon rendeléseket adhatnak le, ehhez először regisztráció szükséges. Ilyenkor a rendeléshez szükséges adataik tárolásra kerülnek. Egy regisztrált felhasználó szabadon adhat le megrendelést. A rendelésnél az ügyfél kiválaszthat különféle fizetési módokat, illetve választhat a különféle szállító cégek közül, melyek adatai szerepelnek a rendszerben. Minden szállítóhoz tartozik egy szállítási ár, ez a rendelés összegéhez adódik hozzá. A rendeléseknek van továbbá státusza, amely lehet „aktív”, „teljesítve”, illetve „törölve”.

A termékekhez nem tartozik sok adat. Tárolásra kerül a termék neve, azonosítója, hogy akciós-e éppen, illetve az ára.

Ha egy, az oldalon szereplő terméket megrendelnek, akkor létrejön a rendszerben egy új elem, amely a rendelt termék. Itt kerül tárolásra a rendeléshez tartozó áfa, a rendelt termék egységára, illetve a megrendelt termék mennyisége. Itt kerül tárolásra az is, hogy a rendelt termék melyik rendeléshez tartozik, egy rendelt termék nem tartozhat több megrendeléshez, viszont egy rendeléshez értelemszerűen több rendelt termék is tartozhat. A rendeléshez tartozó rendelt termékek egységára, szorozva azt a rendelt termékek mennyiségével hozzáadódik a rendelés összegéhez. Így tehát egy rendelés összegét a rendelt termékek árának felszorzásával, summázásával majd a szállítási ár hozzáadásával kapjuk meg.

1. feladat

1a) Az adatbázis ER modell tervezése

A modellben 5 entitás található, ezek: **ügyfél**, **rendelés**, **termék**, **rendelttermék**, **szállító**.

Három fajta kapcsolat (4db):

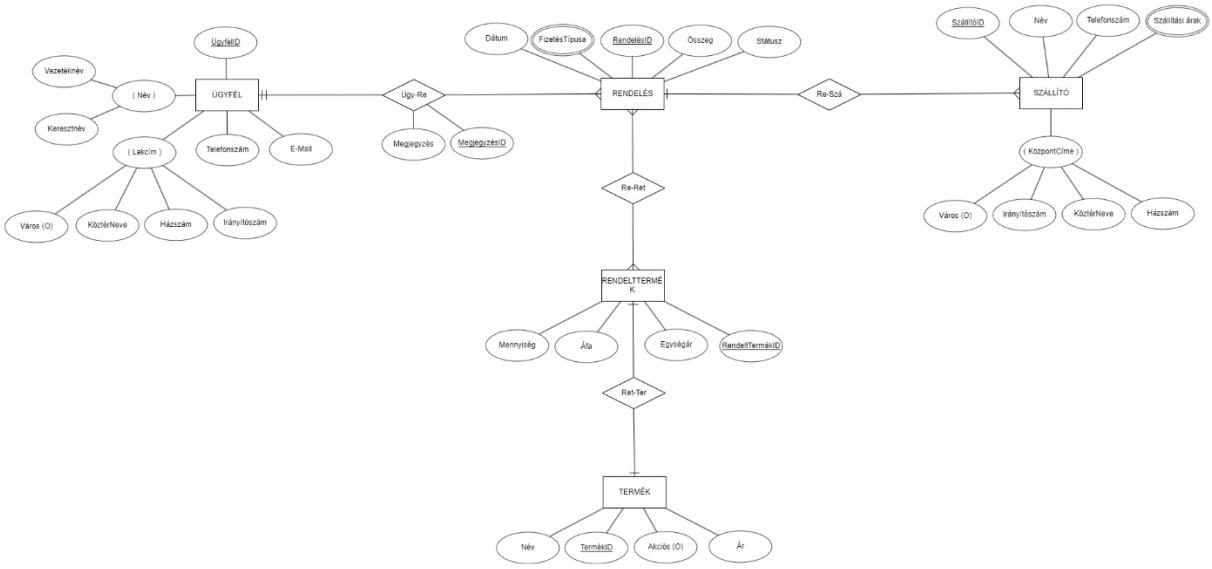
- **1:1** kapcsolat: **Rendelttermék** és **Termék** között (**Ret-Ter)**
- **1:N** kapcsolat: **Ügyfél** és **Rendelés** (**Ügy-Re**), illetve **Rendelés** és **Szállító** között (**Re-Szá)**
- **N:M** kapcsolat: **Rendelés** és **Rendelttermék** között (**Re-Ret)**

A modellben szereplő entitások és tulajdonságaik:

- Ügyfél
    - ÜgyfélID: kulcs (azonosító)
    - (Név): az ügyfél neve
        - Vezetéknév: az ügyfél vezetékneve
        - Keresztnév: az ügyfél keresztneve
    - (Lakcím): az ügyfél lakcíme
        - Irányítószám: a város irányítószáma, amelyben az ügyfél lakik
        - Város(O): a város, amelyben az ügyfél lakik
        - KöztérNeve: az utca, amelyben az ügyfél lakik
        - Házszám: a házszám, ahol az ügyfél lakik
    - Telefonszám: az ügyfél telefonszáma
    - E-Mail: az ügyfél E-mail címe
- Rendelés
    - RendelésID: kulcs (azonosító)
    - Összeg: a rendelés összege
    - FizetésTípusa: a fizetés típusa
    - Státusz: a rendelés státusza
    - Dátum: a rendelés leadásának dátuma
- Rendelttermék
    - RendelttermékID: kulcs (azonosító)
    - Egységár: a rendelt termék egységára
    - Mennyiség: a rendelt termék mennyisége
    - Áfa: a rendelt termékhez (rendeléshez) tartozó áfa

- Termék
    - TermékID: kulcs (azonosító)
    - Név: a termék neve
    - Akciós: megadja, hogy akciós-e a termék
    - Ár: a termék ára
- Szállító
    - SzállítóID: kulcs (azonosító)
    - Név: a szállító cég neve
    - Telefonszám: a szállító cég telefonszáma
    - Szállítási árak: a cég szállítási ára(i)
    - (KözpontCíme):
        - Irányítószám: a város irányítószáma, ahol a cég központja található
        - Város(O): a város, ahol a cég központja található
        - KöztérNeve: az utca, ahol a cég központja található
        - Házszám: a házszám, ahol a cég központja található
- Ügy-Re
    - MegjegyzésID: azonosító (kulcs)
    - Megjegyzés: a rendeléshez tartozó megjegyzés

Az ER modell:

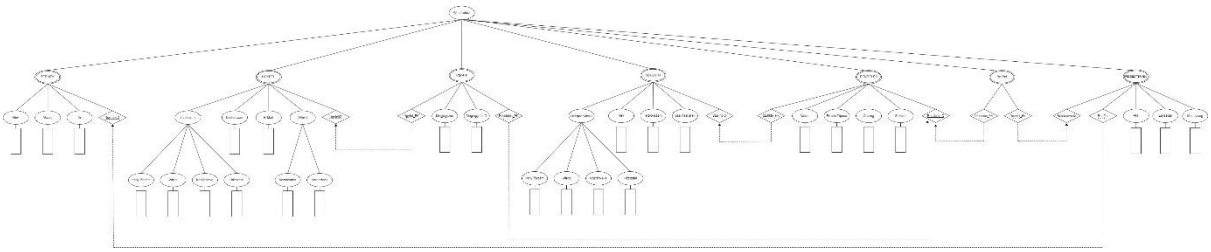


1b) Az adatbázis konvertálása XDM modellre

Az XDM modellt az ER modell alapján készítettem el. A gyökérelemből további 7 darab gyerekelem indul ki, ebből 5 darab entitást-, 2 darab kapcsolatot ír le.

A kapcsolatokat különféle módokon valósítottam meg az alapján az ER modell alapján. Ahol szükséges volt létrehoztam egy új elemet, megadva nekik az idegen kulcsot, és összekapcsoltam őket a megfelelő kulcsokkal. Ahol erre nem volt szükség, ott az elemnél hoztam létre egy idegen kulcsot, melyet összekötöttem a másik elemben lévő azonosítóval.

Az XDM modell:



1c) Az XDM modell alapján XML dokumentum készítése:

A modellek alapján elkészítettem az XML dokumentumot. A dokumentumban a modellek elemei szerepelnek. Az elemek elementek formájában kerültek létrehozásra, bizonyos tulajdonságok pedig az elemek attribútumaiként (például azonosítók), vagy pedig a szülőelemek gyerekelemeként jöttek létre. A dokumentumban kommentekkel jelölöm, hogy éppen melyik elem példányai szerepelnek az adott részben.

A szülő- és gyerekelemek egy az XML-nek megfelelő struktúrát írnak le.

Az XML dokumentum:

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

<webshop xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
xs:noNamespaceSchemaLocation="XMLTaskUY5E1L.xsd">

    <!-- Ügyfelek -->

    <ugyfel ugyfelid="1">
        <nev>
            <vezeteknev>Kovács</vezeteknev>
            <keresztnev>Anna</keresztnev>
        </nev>
        <lakcim>
            <iranyitoszam>1111</iranyitoszam>
            <varos>Budapest</varos>
            <kozterneve>Kossuth út</kozterneve>
            <hazszam>12</hazszam>
        </lakcim>
        <telefonszam>+36301234567</telefonszam>
        <email>anna.kovacs@example.com</email>
    </ugyfel>

    <ugyfel ugyfelid="2">
        <nev>
            <vezeteknev>Nagy</vezeteknev>
            <keresztnev>Péter</keresztnev>
        </nev>
        <lakcim>
            <iranyitoszam>2222</iranyitoszam>
            <varos>Debrecen</varos>
            <kozterneve>Szabadság út</kozterneve>
            <hazszam>34</hazszam>
        </lakcim>
        <telefonszam>+36209876543</telefonszam>
        <email>peter.nagy@example.com</email>
```

```
    </ugyfel>

    <ugyfel ugyfelid="3">
        <nev>
            <vezeteknev>Szabó</vezeteknev>
            <keresztnev>Géza</keresztnev>
        </nev>
        <lakcim>
            <iranyitoszam>3333</iranyitoszam>
            <varos>Szeged</varos>
            <kozterneve>Petőfi út</kozterneve>
            <hazszam>45</hazszam>
        </lakcim>
        <telefonszam>+36705557890</telefonszam>
        <email>geza.szabo@example.com</email>
    </ugyfel>

    <ugyfel ugyfelid="4">
        <nev>
            <vezeteknev>Kiss</vezeteknev>
            <keresztnev>Mária</keresztnev>
        </nev>
        <lakcim>
            <iranyitoszam>4444</iranyitoszam>
            <varos>Pécs</varos>
            <kozterneve>Rákóczi út</kozterneve>
            <hazszam>78</hazszam>
        </lakcim>
        <telefonszam>+36302221111</telefonszam>
        <email>maria.kiss@example.com</email>
    </ugyfel>

    <ugyfel ugyfelid="5">
        <nev>
            <vezeteknev>Varga</vezeteknev>
            <keresztnev>Attila</keresztnev>
        </nev>
        <lakcim>
            <iranyitoszam>5555</iranyitoszam>
            <varos>Győr</varos>
            <kozterneve>Ady út</kozterneve>
            <hazszam>56</hazszam>
        </lakcim>
        <telefonszam>+36203339999</telefonszam>
        <email>attila.varga@example.com</email>
    </ugyfel>

    <ugyfel ugyfelid="6">
```

```
        <nev>
            <vezeteknev>Molnár</vezeteknev>
            <keresztnev>Éva</keresztnev>
        </nev>
        <lakcim>
            <iranyitoszam>6666</iranyitoszam>
            <varos>Szombathely</varos>
            <kozterneve>Arany út</kozterneve>
            <hazszam>89</hazszam>
        </lakcim>
        <telefonszam>+36704445678</telefonszam>
        <email>eva.molnar@example.com</email>
    </ugyfel>

    <ugyfel ugyfelid="7">
        <nev>
            <vezeteknev>Bálint</vezeteknev>
            <keresztnev>József</keresztnev>
        </nev>
        <lakcim>
            <iranyitoszam>7777</iranyitoszam>
            <varos>Miskolc</varos>
            <kozterneve>Dózsa út</kozterneve>
            <hazszam>23</hazszam>
        </lakcim>
        <telefonszam>+36301112222</telefonszam>
        <email>jozsef.balint@example.com</email>
    </ugyfel>

    <ugyfel ugyfelid="8">
        <nev>
            <vezeteknev>Takács</vezeteknev>
            <keresztnev>Katalin</keresztnev>
        </nev>
        <lakcim>
            <iranyitoszam>8888</iranyitoszam>
            <varos>Eger</varos>
            <kozterneve>Széchenyi út</kozterneve>
            <hazszam>67</hazszam>
        </lakcim>
        <telefonszam>+36205553333</telefonszam>
        <email>katalin.takacs@example.com</email>
    </ugyfel>

    <ugyfel ugyfelid="9">
        <nev>
            <vezeteknev>Horváth</vezeteknev>
            <keresztnev>András</keresztnev>
```

```
        </nev>
        <lakcim>
            <iranyitoszam>9999</iranyitoszam>
            <varos>Veszprém</varos>
            <kozterneve>Petőfi út</kozterneve>
            <hazszam>78</hazszam>
        </lakcim>
        <telefonszam>+36708889999</telefonszam>
        <email>andras.horvath@example.com</email>
    </ugyfel>

    <ugyfel ugyfelid="10">
        <nev>
            <vezeteknev>Fekete</vezeteknev>
            <keresztnev>Renáta</keresztnev>
        </nev>
        <lakcim>
            <iranyitoszam>1010</iranyitoszam>
            <varos>Székesfehérvár</varos>
            <kozterneve>Bajnai út</kozterneve>
            <hazszam>34</hazszam>
        </lakcim>
        <telefonszam>+36302224444</telefonszam>
        <email>renata.fekete@example.com</email>
    </ugyfel>

    <ugyfel ugyfelid="11">
        <nev>
            <vezeteknev>Balogh</vezeteknev>
            <keresztnev>Zoltán</keresztnev>
        </nev>
        <lakcim>
            <iranyitoszam>1111</iranyitoszam>
            <varos>Kecskemét</varos>
            <kozterneve>Szabadság út</kozterneve>
            <hazszam>56</hazszam>
        </lakcim>
        <telefonszam>+36207778888</telefonszam>
        <email>zoltan.balogh@example.com</email>
    </ugyfel>

    <ugyfel ugyfelid="12">
        <nev>
            <vezeteknev>Simon</vezeteknev>
            <keresztnev>Krisztina</keresztnev>
        </nev>
        <lakcim>
            <iranyitoszam>1212</iranyitoszam>
```

```
            <varos>Sopron</varos>
            <kozterneve>Rákóczi út</kozterneve>
            <hazszam>12</hazszam>
        </lakcim>
        <telefonszam>+36703335555</telefonszam>
        <email>krisztina.simon@example.com</email>
    </ugyfel>

    <ugyfel ugyfelid="13">
        <nev>
            <vezeteknev>Tóth</vezeteknev>
            <keresztnev>Gábor</keresztnev>
        </nev>
        <lakcim>
            <iranyitoszam>1313</iranyitoszam>
            <varos>Nyíregyháza</varos>
            <kozterneve>Kossuth út</kozterneve>
            <hazszam>45</hazszam>
        </lakcim>
        <telefonszam>+36304446666</telefonszam>
        <email>gabor.toth@example.com</email>
    </ugyfel>

    <ugyfel ugyfelid="14">
        <nev>
            <vezeteknev>Barta</vezeteknev>
            <keresztnev>Enikő</keresztnev>
        </nev>
        <lakcim>
            <iranyitoszam>1414</iranyitoszam>
            <varos>Szolnok</varos>
            <kozterneve>Deák út</kozterneve>
            <hazszam>78</hazszam>
        </lakcim>
        <telefonszam>+36209991111</telefonszam>
        <email>eniko.barta@example.com</email>
    </ugyfel>

    <ugyfel ugyfelid="15">
        <nev>
            <vezeteknev>Király</vezeteknev>
            <keresztnev>István</keresztnev>
        </nev>
        <lakcim>
            <iranyitoszam>1515</iranyitoszam>
            <varos>Pécsvárad</varos>
            <kozterneve>Széchenyi út</kozterneve>
            <hazszam>34</hazszam>
```

```
        </lakcim>
        <telefonszam>+36708882222</telefonszam>
        <email>istvan.kiraly@example.com</email>
    </ugyfel>

    <!-- Ügy-Re -->

    <ugyre megjegyzesid="1" ugyfelid="1" rendelesid="10">
        <megjegyzes>Kérem, minél hamarabb szállítsák ki a
terméket!</megjegyzes>
    </ugyre>

    <ugyre megjegyzesid="2" ugyfelid="8" rendelesid="15">
        <megjegyzes>Csak 12:00-tól vagyok otthon.</megjegyzes>
    </ugyre>

    <ugyre megjegyzesid="3" ugyfelid="14" rendelesid="16">
        <megjegyzes>A kapucsengő kódja 10 kulcs 8521.</megjegyzes>
    </ugyre>

    <ugyre megjegyzesid="4" ugyfelid="9" rendelesid="8">
        <megjegyzes>Jövő hét keddig nem leszek otthon.</megjegyzes>
    </ugyre>

    <!-- Rendelések -->

    <rendeles rendelesid="1" statusz="fizetésre vár" fizetestipusa="online"
ugyfelid="11" szallitoid="3">
        <osszeg>7067.46</osszeg>
        <datum>2023-03-15</datum>
    </rendeles>

    <rendeles rendelesid="2" statusz="törölve" fizetestipusa="online"
ugyfelid="5" szallitoid="3">
        <osszeg>1449.99</osszeg>
        <datum>2023-03-16</datum>
    </rendeles>

    <rendeles rendelesid="3" statusz="aktív" fizetestipusa="utánvét"
ugyfelid="2" szallitoid="2">
        <osszeg>1947.85</osszeg>
        <datum>2023-03-17</datum>
    </rendeles>

    <rendeles rendelesid="4" statusz="teljesítve" fizetestipusa="online"
ugyfelid="4" szallitoid="3">
        <osszeg>2449.99</osszeg>
        <datum>2023-03-18</datum>
```

```
    </rendeles>

    <rendeles rendelesid="5" statusz="teljesítve" fizetestipusa="utánvét"
ugyfelid="5" szallitoid="2">
        <osszeg>8298</osszeg>
        <datum>2023-03-19</datum>
    </rendeles>

    <rendeles rendelesid="6" statusz="aktív" fizetestipusa="utánvét"
ugyfelid="1" szallitoid="2">
        <osszeg>959.98</osszeg>
        <datum>2023-03-20</datum>
    </rendeles>

    <rendeles rendelesid="7" statusz="aktív" fizetestipusa="utánvét"
ugyfelid="15" szallitoid="1">
        <osszeg>6998.45</osszeg>
        <datum>2023-03-21</datum>
    </rendeles>

    <rendeles rendelesid="8" statusz="fizetésre vár" fizetestipusa="online"
ugyfelid="9" szallitoid="1">
        <osszeg>1499</osszeg>
        <datum>2023-03-22</datum>
    </rendeles>

    <rendeles rendelesid="9" statusz="teljesítve" fizetestipusa="online"
ugyfelid="5" szallitoid="3">
        <osszeg>1199.97</osszeg>
        <datum>2023-03-23</datum>
    </rendeles>

    <rendeles rendelesid="10" statusz="aktív" fizetestipusa="utánvét"
ugyfelid="1" szallitoid="2">
        <osszeg>8456.9</osszeg>
        <datum>2023-03-24</datum>
    </rendeles>

    <!-- Rendelt Termékek -->

    <rendelttermek rendelttermekid="1" termekid="3" rendelesid="5">
        <egysegar>1899.50</egysegar>
        <afa>27%</afa>
        <mennyiseg>4</mennyiseg>
    </rendelttermek>

    <rendelttermek rendelttermekid="2" termekid="9" rendelesid="8">
        <egysegar>499.50</egysegar>
```

```
        <afa>27%</afa>
        <mennyiseg>2</mennyiseg>
    </rendelttermek>

    <rendelttermek rendelttermekid="3" termekid="3" rendelesid="1">
        <egysegar>1899.50</egysegar>
        <afa>27%</afa>
        <mennyiseg>3</mennyiseg>
    </rendelttermek>

    <rendelttermek rendelttermekid="4" termekid="12" rendelesid="3">
        <egysegar>149.95</egysegar>
        <afa>27%</afa>
        <mennyiseg>3</mennyiseg>
    </rendelttermek>

    <rendelttermek rendelttermekid="5" termekid="2" rendelesid="2">
        <egysegar>999.99</egysegar>
        <afa>27%</afa>
        <mennyiseg>1</mennyiseg>
    </rendelttermek>

    <rendelttermek rendelttermekid="6" termekid="11" rendelesid="4">
        <egysegar>1999.99</egysegar>
        <afa>27%</afa>
        <mennyiseg>1</mennyiseg>
    </rendelttermek>

    <rendelttermek rendelttermekid="7" termekid="10" rendelesid="1">
        <egysegar>399.00</egysegar>
        <afa>27%</afa>
        <mennyiseg>1</mennyiseg>
    </rendelttermek>

    <rendelttermek rendelttermekid="8" termekid="8" rendelesid="1">
        <egysegar>129.99</egysegar>
        <afa>27%</afa>
        <mennyiseg>4</mennyiseg>
    </rendelttermek>

    <rendelttermek rendelttermekid="9" termekid="7" rendelesid="10">
        <egysegar>1899.00</egysegar>
        <afa>27%</afa>
        <mennyiseg>1</mennyiseg>
    </rendelttermek>

    <rendelttermek rendelttermekid="10" termekid="1" rendelesid="10">
        <egysegar>2799.00</egysegar>
```

```
        <afa>27%</afa>
        <mennyiseg>2</mennyiseg>
    </rendelttermek>

    <rendelttermek rendelttermekid="11" termekid="8" rendelesid="6">
        <egysegar>129.99</egysegar>
        <afa>27%</afa>
        <mennyiseg>2</mennyiseg>
    </rendelttermek>

    <rendelttermek rendelttermekid="12" termekid="5" rendelesid="9">
        <egysegar>249.99</egysegar>
        <afa>27%</afa>
        <mennyiseg>3</mennyiseg>
    </rendelttermek>

    <rendelttermek rendelttermekid="13" termekid="2" rendelesid="7">
        <egysegar>999.99</egysegar>
        <afa>27%</afa>
        <mennyiseg>5</mennyiseg>
    </rendelttermek>

    <rendelttermek rendelttermekid="14" termekid="9" rendelesid="7">
        <egysegar>499.50</egysegar>
        <afa>27%</afa>
        <mennyiseg>3</mennyiseg>
    </rendelttermek>

    <rendelttermek rendelttermekid="15" termekid="6" rendelesid="10">
        <egysegar>129.95</egysegar>
        <afa>27%</afa>
        <mennyiseg>2</mennyiseg>
    </rendelttermek>

    <rendelttermek rendelttermekid="16" termekid="10" rendelesid="3">
        <egysegar>399.00</egysegar>
        <afa>27%</afa>
        <mennyiseg>2</mennyiseg>
    </rendelttermek>

    <!-- Termékek -->

    <termek termekid="1" akcios="nem">
        <nev>Laptop Dell XPS 13</nev>
        <ar>2799.99</ar>
    </termek>

    <termek termekid="2" akcios="nem">
```

```
        <nev>Okostelefon Samsung Galaxy S21</nev>
        <ar>999.99</ar>
    </termek>

    <termek termekid="3" akcios="nem">
        <nev>Televízió LG OLED55CX</nev>
        <ar>1899.50</ar>
    </termek>

    <termek termekid="4" akcios="igen">
        <nev>Fényképezőgép Canon EOS R5</nev>
        <ar>3499.00</ar>
    </termek>

    <termek termekid="5" akcios="nem">
        <nev>Hangszóró Bose SoundLink Revolve+</nev>
        <ar>249.99</ar>
    </termek>

    <termek termekid="6" akcios="igen">
        <nev>Sportcipő Nike Air Zoom Pegasus 38</nev>
        <ar>129.95</ar>
    </termek>

    <termek termekid="7" akcios="nem">
        <nev>Laptop ASUS ROG Zephyrus G14</nev>
        <ar>1899.00</ar>
    </termek>

    <termek termekid="8" akcios="igen">
        <nev>Fitness Tracker Fitbit Charge 5</nev>
        <ar>129.99</ar>
    </termek>

    <termek termekid="9" akcios="nem">
        <nev>Konyhai Robotgép KitchenAid Artisan</nev>
        <ar>499.50</ar>
    </termek>

    <termek termekid="10" akcios="igen">
        <nev>Okosóra Apple Watch Series 7</nev>
        <ar>399.00</ar>
    </termek>

    <termek termekid="11" akcios="nem">
        <nev>Digitális fényképezőgép Sony Alpha A7 III</nev>
        <ar>1999.99</ar>
    </termek>
```

```xml
    <termek termekid="12" akcios="igen">
        <nev>Hangfal JBL Charge 4</nev>
        <ar>149.95</ar>
    </termek>

    <!-- Szállítók -->

    <szallito szallitoid="1">
        <nev>Express Shipping Kft.</nev>
        <telefonszam>+36301234567</telefonszam>
        <szallitasiar>500.00</szallitasiar>
        <kozpontcime>
            <iranyitoszam>1055</iranyitoszam>
            <varos>Budapest</varos>
            <kozterneve>Váci út</kozterneve>
            <hazszam>22</hazszam>
        </kozpontcime>
    </szallito>

    <szallito szallitoid="2">
        <nev>Fast Logistics Zrt.</nev>
        <telefonszam>+36209876543</telefonszam>
        <szallitasiar>700.00</szallitasiar>
        <kozpontcime>
            <iranyitoszam>1132</iranyitoszam>
            <varos>Budapest</varos>
            <kozterneve>Rákospatak út</kozterneve>
            <hazszam>8</hazszam>
        </kozpontcime>
    </szallito>

    <szallito szallitoid="3">
        <nev>Rapid Delivery Services Kft.</nev>
        <telefonszam>+36705557890</telefonszam>
        <szallitasiar>450.00</szallitasiar>
        <kozpontcime>
            <iranyitoszam>1097</iranyitoszam>
            <varos>Budapest</varos>
            <kozterneve>Nagyvárad út</kozterneve>
            <hazszam>14</hazszam>
        </kozpontcime>
    </szallito>

</webshop>
```

1d) Az XML dokumentum alapján XMLSchema

Az XML dokumentum alapján elkészítettem az XMLSchema (XSD) sémát. A séma elején megtalálhatóak az általam létrehozott saját típusok, illetve az elemek a saját típusok után lettek létrehozva. A struktúra megadásakor ezekre referenciaként hivatkoznak az elemek. Kommentekkel jelzem, hogy melyik részen melyik elemmel kapcsolatos dolgok vannak.

A kulcsokat és az azokhoz tartozó referenciákat (idegen kulcsokat) ott helyeztem el, ahová tartoznak.

Az XMLSchema:

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

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

    <!-- Saját típusok (Attribútum) -->

    <xs:simpleType name="statuszTipus">
        <xs:restriction base="xs:string">
            <xs:enumeration value="aktív" />
            <xs:enumeration value="fizetésre vár" />
            <xs:enumeration value="teljesítve" />
            <xs:enumeration value="törölve" />
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="fizetesTipus">
        <xs:restriction base="xs:string">
            <xs:enumeration value="online" />
            <xs:enumeration value="utánvét" />
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="akcioTipus">
        <xs:restriction base="xs:string">
            <xs:enumeration value="igen" />
            <xs:enumeration value="nem" />
        </xs:restriction>
    </xs:simpleType>

    <!-- Saját típusok (Element) -->

    <xs:simpleType name="telszamTipus">
        <xs:restriction base="xs:string">
```

```
            <xs:pattern value="\+36\d{9}" />
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="emailTipus">
        <xs:restriction base="xs:string">
            <xs:pattern value="[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-
Z]{2,}"/>
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="afaTipus">
        <xs:restriction base="xs:string">
            <xs:pattern value="\d+%" />
        </xs:restriction>
    </xs:simpleType>

    <!-- Elementek -->

    <!-- Ügyfél (+cím adatok a szállítónál is) -->
    <xs:element name="vezeteknev" type="xs:string" />
    <xs:element name="keresztnev" type="xs:string" />
    <xs:element name="iranyitoszam" type="xs:int" />
    <xs:element name="varos" type="xs:string" />
    <xs:element name="kozterneve" type="xs:string" />
    <xs:element name="hazszam" type="xs:int" />
    <xs:element name="telefonszam" type="telszamTipus" />
    <xs:element name="email" type="emailTipus" />

    <!-- Ügy-Re -->
    <xs:element name="megjegyzes" type="xs:string" />

    <!-- Rendelés -->
    <xs:element name="osszeg" type="xs:float" />
    <xs:element name="datum" type="xs:date" />

    <!-- Rendelt Termék -->
    <xs:element name="egysegar" type="xs:float" />
    <xs:element name="afa" type="afaTipus" />
    <xs:element name="mennyiseg" type="xs:int" />

    <!-- Termék (+a név a szállítónál is) -->
    <xs:element name="nev" type="xs:string" />
    <xs:element name="ar" type="xs:float" />

    <!-- Szállító -->
    <xs:element name="szallitasiar" />
```

```
    <!-- Struktúra megadása -->
    <xs:element name="webshop">
        <xs:complexType>
            <xs:sequence>

                <!-- Ügyfelek -->
                <xs:element name="ugyfel" minOccurs="1" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="nev" minOccurs="1"
maxOccurs="1">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element ref="vezeteknev"
minOccurs="1" maxOccurs="1" />
                                        <xs:element ref="keresztnev"
minOccurs="1" maxOccurs="1" />
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>

                            <xs:element name="lakcim" minOccurs="1"
maxOccurs="1">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element ref="iranyitoszam"
minOccurs="1" maxOccurs="1" />
                                        <xs:element ref="varos" minOccurs="0"
maxOccurs="1" />
                                        <xs:element ref="kozterneve"
minOccurs="1" maxOccurs="1" />
                                        <xs:element ref="hazszam"
minOccurs="1" maxOccurs="1" />
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>

                            <xs:element ref="telefonszam" minOccurs="1"
maxOccurs="1" />
                            <xs:element ref="email" minOccurs="1"
maxOccurs="1" />
                        </xs:sequence>
                        <xs:attribute name="ugyfelid" type="xs:int"
use="required" />
                    </xs:complexType>

                    <!-- Ügyfél ID (kulcs) -->
                    <xs:key name="ugyfel_key">
```

```
                    <xs:selector xpath="ugyfel"></xs:selector>
                    <xs:field xpath="@ugyfelid"></xs:field>
                </xs:key>
            </xs:element>

            <!-- Kapcsolótábla (Ügy-Re) -->
            <xs:element name="ugyre" minOccurs="1" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element ref="megjegyzes" minOccurs="1"
maxOccurs="1" />
                    </xs:sequence>
                    <xs:attribute name="megjegyzesid" type="xs:int"
use="required" />
                    <xs:attribute name="ugyfelid" type="xs:int"
use="required" />
                    <xs:attribute name="rendelesid" type="xs:int"
use="required" />
                </xs:complexType>

                <!-- Megjegyzés ID (kulcs) -->
                <xs:key name="megjegyzes_key">
                    <xs:selector xpath="ugyre"></xs:selector>
                    <xs:field xpath="@megjegyzesid"></xs:field>
                </xs:key>

                <!-- Megjegyzés idegen kulcsok -->
                <xs:keyref name="ugyre_ugyfel_fkey" refer="ugyfel_key">
                    <xs:selector xpath="megjegyzes"></xs:selector>
                    <xs:field xpath="@ugyfelid"></xs:field>
                </xs:keyref>

                <xs:keyref name="ugyre_rendeles_fkey"
refer="rendeles_key">
                    <xs:selector xpath="megjegyzes"></xs:selector>
                    <xs:field xpath="@ugyfelid"></xs:field>
                </xs:keyref>
            </xs:element>

            <!-- Rendelések  -->
            <xs:element name="rendeles" minOccurs="1"
maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element ref="osszeg" minOccurs="1"
maxOccurs="1" />
                        <xs:element ref="datum" minOccurs="1"
maxOccurs="1" />
```

```
                        </xs:sequence>
                        <xs:attribute name="rendelesid" type="xs:int"
use="required" />
                        <xs:attribute name="statusz" type="statuszTipus"
use="required" />
                        <xs:attribute name="fizetestipusa" type="fizetesTipus"
use="required" />
                        <xs:attribute name="ugyfelid" type="xs:int"
use="required" />
                        <xs:attribute name="szallitoid" type="xs:int"
use="required" />
                    </xs:complexType>

                    <!-- Rendelés ID (kulcs) -->
                    <xs:key name="rendeles_key">
                        <xs:selector xpath="rendeles"></xs:selector>
                        <xs:field xpath="@rendelesid"></xs:field>
                    </xs:key>

                    <!-- Rendelés idegen kulcsok -->
                    <xs:keyref name="rendeles_ugyfel_fkey" refer="ugyfel_key">
                        <xs:selector xpath="rendeles"></xs:selector>
                        <xs:field xpath="@ugyfelid"></xs:field>
                    </xs:keyref>

                    <xs:keyref name="rendeles_szallito_fkey"
refer="szallito_key">
                        <xs:selector xpath="rendeles"></xs:selector>
                        <xs:field xpath="@szallitoid"></xs:field>
                    </xs:keyref>
                </xs:element>

                <!-- Rendelt Termékek -->
                <xs:element name="rendelttermek" minOccurs="1"
maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element ref="egysegar" minOccurs="1"
maxOccurs="1" />
                            <xs:element ref="afa" minOccurs="1" maxOccurs="1"
/>
                            <xs:element ref="mennyiseg" maxOccurs="1" />
                        </xs:sequence>
                        <xs:attribute name="rendelttermekid" type="xs:int"
use="required" />
                        <xs:attribute name="termekid" type="xs:int"
use="required" />
```

```
                        <xs:attribute name="rendelesid" type="xs:int"
use="required" />
                    </xs:complexType>

                    <!-- Rendelés ID (kulcs) -->
                    <xs:key name="rendelttermek_key">
                        <xs:selector xpath="rendelttermek"></xs:selector>
                        <xs:field xpath="@rendelttermekid"></xs:field>
                    </xs:key>

                    <!-- Rendelt Termék idegen kulcsok -->
                    <xs:keyref name="rendelttermek_termek_fkey"
refer="termek_key">
                        <xs:selector xpath="rendelttermek"></xs:selector>
                        <xs:field xpath="@termekid"></xs:field>
                    </xs:keyref>

                    <xs:keyref name="rendelttermek_rendeles_fkey"
refer="rendeles_key">
                        <xs:selector xpath="rendelttermek"></xs:selector>
                        <xs:field xpath="@rendelesid"></xs:field>
                    </xs:keyref>
                </xs:element>

                <!-- Termékek -->
                <xs:element name="termek" minOccurs="1" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element ref="nev" minOccurs="1" maxOccurs="1"
/>
                            <xs:element ref="ar" minOccurs="1" maxOccurs="1"
/>
                        </xs:sequence>
                        <xs:attribute name="termekid" type="xs:int"
use="required" />
                        <xs:attribute name="akcios" type="akcioTipus"
use="required" />
                    </xs:complexType>

                    <!-- Termék ID (kulcs) -->
                    <xs:key name="termek_key">
                        <xs:selector xpath="termek"></xs:selector>
                        <xs:field xpath="@termekid"></xs:field>
                    </xs:key>
                </xs:element>

                <!-- Szállító -->
```

```
                <xs:element name="szallito" minOccurs="1"
maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element ref="nev" minOccurs="1" maxOccurs="1"
/>
                            <xs:element ref="telefonszam" minOccurs="1"
maxOccurs="1" />
                            <xs:element ref="szallitasiar" minOccurs="1"
maxOccurs="1" />
                            <xs:element name="kozpontcime">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element ref="iranyitoszam"
minOccurs="1" maxOccurs="1" />
                                        <xs:element ref="varos" minOccurs="0"
maxOccurs="1" />
                                        <xs:element ref="kozterneve"
minOccurs="1" maxOccurs="1" />
                                        <xs:element ref="hazszam"
minOccurs="1" maxOccurs="1" />
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                        <xs:attribute name="szallitoid" type="xs:int"
use="required" />
                    </xs:complexType>

                    <!-- Szállító ID (kulcs) -->
                    <xs:key name="szallito_key">
                        <xs:selector xpath="szallito"></xs:selector>
                        <xs:field xpath="@szallitoid"></xs:field>
                    </xs:key>
                </xs:element>

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

</xs:schema>
```

2. feladat

2a) Adatolvasás

A feladat elkészítéséhez beolvastam az XML dokumentumot, majd elkezdtem felépíteni a struktúrát. Először is megadtam az XML-t, majd a gyökérelemet. Aztán a gyökérelem alapján a további gyerekelemeket, majd azok gyerekelemeit egy rekurzívan, hasonlóan működő metódussal valósítottam meg. A struktúra megtartásához behúzásokat használtam, ehhez külön „indent” és „indentStr” változókat hoztam létre. Az így létrejövő elemeket egy string listába mentettem, majd ebből kerültek kiíratásra a konzolra és az új fájlba.

A kód:

```
package hu.domparse.uy5e1l;

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

import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.parsers.DocumentBuilder;

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

public class DOMReadUY5E1L {

    //A sorok listája
    public static List<String> allLines = new ArrayList<>();

    public static void main(String[] args) {
        try {
            //Fájl beolvasása
            File inputFile = new
File("src/hu/domparse/uy5e1l/XMLTaskUY5E1L.xml");

            //DOM létrehozása
```

```
            DocumentBuilderFactory dbFactory =
DocumentBuilderFactory.newInstance();
            DocumentBuilder domBuilder = dbFactory.newDocumentBuilder();
            Document doc = domBuilder.parse(inputFile);

            //XML
            String xml = "<?xml version=\"1.0\" encoding=\"UTF-8\"?>" +
"\n".repeat(2);
            allLines.add(xml);

            //Gyökér element meghatározása
            Node root = doc.getDocumentElement();
            allLines.add(getRoot(root));

            //Gyermek elementek listájának létrehozása
            NodeList childNodeList = root.getChildNodes();

            //Behúzás megvalósítása
            int indent = 1;
            String indentStr = "\t";

            //Dokumentum elemeinek beolvasása
            for(int i = 0; i < childNodeList.getLength(); i++) {
                Node childNode = childNodeList.item(i);
                if(childNode.getNodeType() == Node.ELEMENT_NODE) {
                    Element childElement = (Element) childNode;

                    allLines.add("\n" + indentStr.repeat(indent) + "<" +
childElement.getNodeName() + getAttributes(childElement) + ">");
                    getChildNodes(childElement, indent, indentStr);
                    allLines.add("\n" + indentStr.repeat(indent) + "</" +
childElement.getNodeName() + ">\n");
                }
            }

            //Gyökér element lezárása a dokumentum végén
            allLines.add("\n"+ "</" + root.getNodeName() + ">");

            //Kiíratás consolre-ra és fájlba
            try {

                //Létrehozom a FileWritert
                FileWriter writer = new
FileWriter("src/hu/domparse/uy5e1l/XMLTaskUY5E1LOutput.xml");

                //Létrehozom a BufferedWritert a FileWriterre
                BufferedWriter buffWriter = new BufferedWriter(writer);
```

```
                //A listából a ciklussal kiíratom a képernyőre a sorokat, majd
fájlba
                for(String line : allLines) {
                    System.out.print(line);
                    buffWriter.write(line);
                    buffWriter.flush();
                }

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

    public static void getChildNodes(Element parentElement, int indent, String
indentStr) {
        if(parentElement.hasChildNodes()) {
            NodeList childNodeList = parentElement.getChildNodes();

            for(int i = 0; i < childNodeList.getLength(); i++) {
                Node childNode = childNodeList.item(i);
                if(childNode.getNodeType() == Node.ELEMENT_NODE) {
                    Element childElement = (Element) childNode;
                    indent++;

                    //Ellenőrzés, hogy a childElement node-jai közt van-e
element
                    boolean hasInnerElements = false;

                    NodeList checkNodeList = childElement.getChildNodes();

                    for(int j = 0; j < checkNodeList.getLength(); j++) {
                        Node checkNode = checkNodeList.item(j);
                        if(checkNode.getNodeType() == Node.ELEMENT_NODE) {
                            hasInnerElements = true;
                        }
                    }

                    //Hozzáadja a sort a listához, annak függvényében, hogy az
elementnek van-e gyermek elementje, vagy nincs
                    if(hasInnerElements) {
                        allLines.add("\n" + indentStr.repeat(indent) + "<" +
childElement.getNodeName() + getAttributes(childElement) + ">");
                        getChildNodes(childElement, indent, indentStr);
```

```
                        allLines.add("\n" + indentStr.repeat(indent) + "</" +
childElement.getNodeName() + ">");
                    } else {
                        allLines.add("\n"+ indentStr.repeat(indent) + "<" +
childElement.getNodeName() + getAttributes(childElement) + ">");

                        if(childElement.getTextContent().trim().length() > 0)
                            allLines.add(childElement.getTextContent());

                        allLines.add("</" + childElement.getNodeName() + ">");
                    }

                    indent--;
                }
            }
        }
    }

    //Elementek attribútumainak lekérése
    public static String getAttributes(Element parentElement) {
        NamedNodeMap attributes = parentElement.getAttributes();
        String attributeReturn = "";

        for(int i = 0; i < attributes.getLength(); i++) {
            Attr attribute = (Attr) attributes.item(i);

            attributeReturn += " " + attribute;
        }

        return attributeReturn;
    }

    //A gyökér element lekérdezése
    public static String getRoot(Node root){
        String rootReturn = "<" + root.getNodeName();
        NamedNodeMap rootElement = root.getAttributes();

        for(int i = 0; i < rootElement.getLength(); i++){
            Attr attribute = (Attr) rootElement.item(i);
            rootReturn += " " + attribute;
        }

        rootReturn += ">\n";

        return rootReturn;
    }
}
```

2b) Adatmódosítás

Az XML dokumentumban 5 módosítást végeztem el, ezek:

- Az 5-ös azonosítójú termék árának megváltoztatása
- A 10-es azonosítójú ügyfél keresztnevének megváltoztatása
- A 7-es azonosítójú rendelés státuszának megváltoztatása
- A 16-os azonosítójú rendelt termék mennyiségének megváltoztatása
- A 2-es azonosítójú szállító telefonszámának megváltoztatása

Majd a módosított dokumentumot egy „modositott_xml.xml” nevű fájlba kiírattam.

A kód:

```
package hu.domparse.uy5e1l;

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

import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
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 javax.xml.parsers.DocumentBuilder;

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

public class DOMModifyUY5E1L {

    public static void main(String[] args) {

        File inputFile = new File("src/hu/domparse/uy5e1l/XMLTaskUY5E1L.xml");

        try {

            //Document létrehozása
            DocumentBuilderFactory dbFactory =
DocumentBuilderFactory.newInstance();
            DocumentBuilder domBuilder = dbFactory.newDocumentBuilder();
            Document doc = domBuilder.parse(inputFile);

            //Módosítást végző metódus meghívása
            modifyDocument(doc);
```

```
            // Módosított fájl mentése
            TransformerFactory transformerFactory =
TransformerFactory.newInstance();
            Transformer transformer = transformerFactory.newTransformer();
            DOMSource source = new DOMSource(doc);
            StreamResult result = new StreamResult(new
File("src/hu/domparse/uy5e1l/modositott_xml.xml"));
            transformer.transform(source, result);

            //Console-ra való kiíratás
            StreamResult resultConsole = new StreamResult(System.out);
            transformer.transform(source, resultConsole);
        } catch(TransformerException | ParserConfigurationException |
SAXException | IOException e) {
            e.printStackTrace();
        }

    }

    public static void modifyDocument(Document doc) {
        //Egy termék árának megváltoztatása (id 5)
        NodeList productList = doc.getElementsByTagName("termek");
        Element product = (Element) productList.item(4);
        product.getElementsByTagName("ar").item(0).setTextContent("895.99");

        //Egy ügyfél keresztnevének megváltoztatása (id 10)
        NodeList nameList = doc.getElementsByTagName("nev");
        Element name = (Element) nameList.item(9);
        name.getElementsByTagName("keresztnev").item(0).setTextContent("Béla")
;

        //Egy rendelés státuszának megváltoztatása (id 7)
        NodeList orderList = doc.getElementsByTagName("rendeles");
        Element order = (Element) orderList.item(6);
        order.getAttributeNode("statusz").setTextContent("teljesítve");

        //Egy rendelt termék mennyiségének megváltoztatása (id 16)
        NodeList orderedProductList =
doc.getElementsByTagName("rendelttermek");
        Element orderedProduct = (Element) orderedProductList.item(15);
        orderedProduct.getElementsByTagName("mennyiseg").item(0).setTextConten
t("6");

        //Egy szállító telefonszámának megváltoztatása (id 2)
        NodeList shipperList = doc.getElementsByTagName("szallito");
        Element shipper = (Element) shipperList.item(1);
```

```
        shipper.getElementsByTagName("telefonszam").item(0).setTextContent("+3
6708887772");
    }

}
```

2c) Adatlekérdezés

A feladat elkészítésekor 5 lekérdezést hoztam létre. Ezek közül az egyik lekérdezés 2 lekérdezést is megvalósít.

A lekérdezések:

1. A 6-os azonosítójú rendelésben rendelt termék adatainak lekérdezése
2. A 10-es azonosítójú rendelésnél
3. Minden olyan ügyfél adatai, akihez nem tartozik rendelés
4. A szállítók adatai, akik 3-mas azonosítójú terméket szállítanak/szállítottak
5. Minden olyan termék adatai, amelyből már 3-nál több darabot rendeltek

A lekérdezések eredményei a konzolra kerülnek kiíratásra.

A kód:

```
package hu.domparse.uy5e1l;

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

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;

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

public class DOMQueryUY5E1L {

    public static void main(String[] args) {
        try {
            //Fájl beolvasása
            File inputFile = new
File("src/hu/domparse/uy5e1l/XMLTaskUY5E1L.xml");

            //DOM létrehozása
            DocumentBuilderFactory dbFactory =
DocumentBuilderFactory.newInstance();
```

```
            DocumentBuilder domBuilder = dbFactory.newDocumentBuilder();
            Document doc = domBuilder.parse(inputFile);

            //A lekérdezésekhez szükséges NodeList-ek létrehozása
            NodeList orderList = doc.getElementsByTagName("rendeles");
            NodeList orderedProductList =
doc.getElementsByTagName("rendelttermek");
            NodeList productList = doc.getElementsByTagName("termek");
            NodeList ugyreList = doc.getElementsByTagName("ugyre");
            NodeList customerList = doc.getElementsByTagName("ugyfel");
            NodeList shipperList = doc.getElementsByTagName("szallito");

            //A 6-os ID-jű rendelésben rendelt termék adatainak lekérdezése

            System.out.println("1. Lekérdezés:");
            System.out.println("--------------");

            //A keresett rendelés azonósítója
            String searchedOrderID = "6";

            //Először megkeressük a megfelelő azonosítójú rendelést
            for(int i = 0; i < orderList.getLength(); i++) {
                Element order = (Element) orderList.item(i);
                String orderID = order.getAttribute("rendelesid");

                //Ellenőrizzük, hogy a rendelés azonosítója megegyezik-e az
általunk megadott értékkel
                if(searchedOrderID.equals(orderID)) {

                    /* Ha megegyezik, akkor végigmegyünk a rendelt termékek
listáján keresve benne azt az elemet,
                     * amelyben a megfelelő azonosítójú rendelés kulcsa
szerepel
                    */
                    for(int j = 0; j < orderedProductList.getLength(); j++) {
                        Element orderedProduct = (Element)
orderedProductList.item(j);
                        String orderedProductOrderID =
orderedProduct.getAttribute("rendelesid");
                        String productFkey =
orderedProduct.getAttribute("termekid");

                        //Ellenőrizzük, hogy a rendelt termékben szereplő
idegen kulcs megegyezik-e a rendelés kulcsával
                        if(orderedProductOrderID.equals(orderID)) {
                            /* Ha megegyezik, akkor végig megyünk a termékek
listáján keresve, hogy a rendelt terméknél
```

```
                             * szereplő termékazonosító melyik termék
azonosítója
                            */
                            for(int k = 0; k < productList.getLength(); k++) {
                                Element product = (Element)
productList.item(k);
                                String productID =
product.getAttribute("termekid");

                                /* Ellenőrizzük, hogy az éppen "vizsgált"
termék azonosítója megegyezik-e a rendelt termékben
                                 * szereplő kulccsal, majd ha egyezést
találunk, akkor kiíratjuk a termék adatait
                                 */
                                if(productID.equals(productFkey)) {
                                    System.out.println("A rendelésben szereplő
termék");
                                    System.out.println("\tNeve: " +
product.getElementsByTagName("nev").item(0).getTextContent());
                                    System.out.println("\tÁra: " +
product.getElementsByTagName("ar").item(0).getTextContent());
                                }
                            }
                        }
                    }
                }
            }

            System.out.println();

            //A 10-es ID-jű rendelésnél az ügyfél-rendelés kapcsolat adatainak
lekérdezése (megjegyzés, ügyfél adatai)
            System.out.println("2. Lekérdezés:");
            System.out.println("--------------");

            //A keresett rendelés azonósítója
            searchedOrderID = "10";

            //Először megkeressük a megfelelő azonosítójú rendelést
            for(int i = 0; i < orderList.getLength(); i++) {
                Element order = (Element) orderList.item(i);
                String orderID = order.getAttribute("rendelesid");

                //Ellenőrizzük, hogy a rendelés azonosítója megegyezik-e az
általunk megadott értékkel
                if(searchedOrderID.equals(orderID)) {
```

```
                    /* Ha megegyezik, akkor végigmegyünk az ügyfél-rendelés
kapcsolatok listáján keresve benne azt az elemet,
                     * amelyben a rendelés kulcsa megegyezik az általunk
keresett rendelésazonosítóval
                    */
                    for(int j = 0; j < ugyreList.getLength(); j++) {
                        Element ugyre = (Element) ugyreList.item(j);
                        String ugyreOrderID =
ugyre.getAttribute("rendelesid");

                        //Ellenőrizzük, hogy a rendelés azonosítója
megegyezik-e az Ügy-Re kapcsolatban szereplő kulccsal
                        if(orderID.equals(ugyreOrderID)) {

                            //Ha megegyezik, kiíratjuk a benne szereplő
megjegyzést
                            System.out.println("A rendeléshez tartozó
megjegyzés:");
                            System.out.println(ugyre.getElementsByTagName("meg
jegyzes").item(0).getTextContent());
                            String ugyreCustomerID =
ugyre.getAttribute("ugyfelid");

                            /* Ezt követően végigmegyünk az ügyfelek listáján
keresve azt az ügyfelet,
                             * akihez a megjegyzés (illetve a rendelés)
tartozik
                             */
                            for(int k = 0; k < customerList.getLength(); k++)
{
                                Element customer = (Element)
customerList.item(k);
                                String customerID =
customer.getAttribute("ugyfelid");

                                //Ha egyezést találunk, akkor kiíratjuk az
ügyfél adatait
                                if(customerID.equals(ugyreCustomerID)) {
                                    String customerFirstName =
customer.getElementsByTagName("keresztnev").item(0).getTextContent();
                                    String customerLastName =
customer.getElementsByTagName("vezeteknev").item(0).getTextContent();
                                    String postalCode =
customer.getElementsByTagName("iranyitoszam").item(0).getTextContent();
                                    String city =
customer.getElementsByTagName("varos").item(0).getTextContent();
                                    String street =
customer.getElementsByTagName("kozterneve").item(0).getTextContent();
```

```
                                    String houseNumber =
customer.getElementsByTagName("hazszam").item(0).getTextContent();
                                    String phoneNumber =
customer.getElementsByTagName("telefonszam").item(0).getTextContent();
                                    String email =
customer.getElementsByTagName("email").item(0).getTextContent();

                                    System.out.println("\nAz ügyfél adatai,
akihez akihez a rendelés tartozik:");
                                    System.out.println("\tNév: " +
customerLastName + " " + customerFirstName);
                                    System.out.println("\tLakcím: " +
postalCode + ", " + city + ", " + street + " " + houseNumber + ".");
                                    System.out.println("\tTelefonszám: " +
phoneNumber);
                                    System.out.println("\tE-mail: " + email);
                                }
                            }
                        }
                    }
                }
            }

            System.out.println();

            //Minden olyan ügyfél adatai, akihez nem tartozik rendelés
            System.out.println("3. Lekérdezés:");
            System.out.println("--------------");

            //Létrehozzuk a listát, amiben később tároljuk az ügyfelek
azonosítóit
            List<String> customerIDList = new ArrayList<>();

            //Végigmegyünk a rendelések listáján
            for(int i = 0; i < orderList.getLength(); i++) {
                Element order = (Element) orderList.item(i);
                String customerID = order.getAttribute("ugyfelid");

                //Ha egy ügyfél azonosítója még nem szerepel a listában, akkor
hozzáadjuk
                if(!customerIDList.contains(customerID))
                    customerIDList.add(customerID);
            }

            System.out.println("Ügyfelek adatai, akikhez nem tartozik
rendelés:\n");

            int counter = 1;
```

```
            //Végigmegyünk az ügyfelek listáján
            for(int i = 0; i < customerList.getLength(); i++) {
                Element customer = (Element) customerList.item(i);
                String customerID = customer.getAttribute("ugyfelid");


                /* Ellenőrizzük, hogy az éppen "vizsgált" ügyfél azonosítója
szerepel-e a listánkban.
                 * Ha az ügyfél azonosítója nem szerepel a listánkban, az azt
jelenti, hogy nem tartozik
                 * hozzá rendelés, ezért kiíratjuk az adatait.
                 */
                if(!customerIDList.contains(customerID)) {
                    System.out.println(counter + ". Ügyfél (ID:" + customerID
+")");

                    String customerFirstName =
customer.getElementsByTagName("keresztnev").item(0).getTextContent();
                    String customerLastName =
customer.getElementsByTagName("vezeteknev").item(0).getTextContent();
                    String postalCode =
customer.getElementsByTagName("iranyitoszam").item(0).getTextContent();
                    String city =
customer.getElementsByTagName("varos").item(0).getTextContent();
                    String street =
customer.getElementsByTagName("kozterneve").item(0).getTextContent();
                    String houseNumber =
customer.getElementsByTagName("hazszam").item(0).getTextContent();
                    String phoneNumber =
customer.getElementsByTagName("telefonszam").item(0).getTextContent();
                    String email =
customer.getElementsByTagName("email").item(0).getTextContent();

                    System.out.println("\tNév: " + customerLastName + " " +
customerFirstName);
                    System.out.println("\tLakcím: " + postalCode + ", " + city
+ ", " + street + " " + houseNumber + ".");
                    System.out.println("\tTelefonszám: " + phoneNumber);
                    System.out.println("\tE-mail: " + email + "\n");

                    counter++;
                }
            }

            //A szállítók adatai, akik 3-mas idjű terméket
szállítanak/szállítottak
            System.out.println("4. Lekérdezés:");
```

```
            System.out.println("--------------");

            //A keresett termék azonosítója
            String searchedProductID = "3";

            System.out.println("Szállítók, melyek szállítanak/szállítottak 3-
mas idjű terméket:\n");

            counter = 1;

            //Először megkeressük a megfelelő rendelt terméket
            for(int i = 0; i < orderedProductList.getLength(); i++) {
                Element orderedProduct = (Element) orderedProductList.item(i);
                String productFkey = orderedProduct.getAttribute("termekid");
                String orderFkey = orderedProduct.getAttribute("rendelesid");

                //Ellenőrizzük, hogy a rendelt terméknél a termék kulcsa
megegyezik-e az általunk megadott értékkel
                if(productFkey.equals(searchedProductID)) {

                    /* Ha megegyezik, akkor végigmegyünk a rendelések listáján
keresve benne azt az elemet,
                     * amelynél a rendelésazonosító megegyezik a rendelt
terméknél szereplő kulccsal
                    */
                    for(int j = 0; j < orderList.getLength(); j++) {
                        Element order = (Element) orderList.item(j);
                        String orderID = order.getAttribute("rendelesid");
                        String shipperFkey = order.getAttribute("szallitoid");

                        //Ellenőrizzük, hogy a rendelésazonosító megegyezik-e
a rendelt terméknél szereplő kulccsal
                        if(orderID.equals(orderFkey)) {

                            /* Ha megegyezik, akkor végigmegyünk a szállítók
listáján keresve benne azt az elemet,
                             * amelynél a szállító azonosítója megegyezik a
rendelésben szereplő kulccsal
                            */
                            for(int k = 0; k < shipperList.getLength(); k++) {
                                Element shipper = (Element)
shipperList.item(k);
                                String shipperID =
shipper.getAttribute("szallitoid");


                                /* Ellenőrizzük, hogy a szállító azonosítója
megegyezik-e a rendelésben szereplő kulccsal.
```

```
                                * Ha megegyezik, akkor megtaláltuk azt a
szállítót, aki olyan rendelésnek (is) a szállítója,
                                * amelyben szerepel a 3-mas azonosítójú
termék.
                                */
                               if(shipperID.equals(shipperFkey)) {
                                   System.out.println(counter + ". Szállító
(ID:" + shipperID +")");

                                   String name =
shipper.getElementsByTagName("nev").item(0).getTextContent();
                                   String phoneNumber =
shipper.getElementsByTagName("telefonszam").item(0).getTextContent();
                                   String postalCode =
shipper.getElementsByTagName("iranyitoszam").item(0).getTextContent();
                                   String city =
shipper.getElementsByTagName("varos").item(0).getTextContent();
                                   String street =
shipper.getElementsByTagName("kozterneve").item(0).getTextContent();
                                   String houseNumber =
shipper.getElementsByTagName("hazszam").item(0).getTextContent();

                                   System.out.println("\tNév: " + name);
                                   System.out.println("\tKözpont címe: " +
postalCode + ", " + city + ", " + street + " " + houseNumber + ".");
                                   System.out.println("\tTelefonszám: " +
phoneNumber);

                                   counter++;
                               }
                           }
                       }
                   }
               }
           }

           System.out.println();

           //Minden olyan termék adatai, amelyből már 3-nál több darabot
rendeltek
           System.out.println("5. Lekérdezés:");
           System.out.println("--------------");

           /* Létrehozunk egy Map-et, amiben tárolni fogjuk a termékek
azonosítóit, illetve azt,
            * hogy hány darabot rendeltek már belőle
            */
           Map<String, Integer> productAmountMap = new HashMap<>();
```

```
            System.out.println("Termékek, melyekből 3-nál több darabot
rendeltek:\n");

            /* Végigmegyünk a rendelt termékek listáján, majd hozzáadjuk az
ott szereplő termékazonosítókat,
             * illetve hozzájuk tartozó mennyiséget (végül a mennyiségek
összegét) a Map-hez
             */
            for(int i = 0; i < orderedProductList.getLength(); i++) {
                Element orderedProduct = (Element) orderedProductList.item(i);
                String productFkey = orderedProduct.getAttribute("termekid");

                //A termékazonosítót átkonvertáljuk integer-ré, hogy számként
tudjuk kezelni
                int amount =
Integer.parseInt(orderedProduct.getElementsByTagName("mennyiseg").item(0).getT
extContent());

                /* Ellenőrizzük, hogy a Map tartalmazza-e már a rendelt
terméknél szereplő termékazonosítót,
                 * és csak akkor adjuk hozzá a kulcsot és a mennyiséget, ha
még nem szerepel a Map-ben.
                 *
                 * Ha már szerepel, abban az esetben lekérjük a kulcshoz
tartozó értéket (mennyiség) egy változóba,
                 * majd hozzáadjuk a jelenleg lekért mennyiséget a változóhoz,
és az alapján  "frissítjük" az értéket a Map-ben.
                 */
                if(!productAmountMap.containsKey(productFkey)) {
                    productAmountMap.put(productFkey, amount);
                } else {
                    int tempAmount = productAmountMap.get(productFkey);
                    tempAmount += amount;
                    productAmountMap.put(productFkey, tempAmount);
                }
            }

            counter = 1;

            //Végigmegyünk a termékek listáján
            for(int i = 0; i < productList.getLength(); i++) {
                Element product = (Element) productList.item(i);
                String productID = product.getAttribute("termekid");

                /* Ha az éppen vizsgált termék azonosítója szerepel a Map-ben,
illetve teljesül
```

```
                * a feltétel, hogy a hozzátartozó érték (mennyiség) nagyobb,
mint 3, akkor kiíratjuk
                * a termék adatait, illetve a rendelések számát, ami azt
takarja, hogy eddig
                * összesen hány darabot rendeltek a termékből.
                */
               if(productAmountMap.containsKey(productID)) {
                   if(productAmountMap.get(productID) > 3) {
                       System.out.println(counter + ". Termék (ID:" +
productID +", Rendelések száma: " + productAmountMap.get(productID) + ")");
                       System.out.println("\tNeve: " +
product.getElementsByTagName("nev").item(0).getTextContent());
                       System.out.println("\tÁra: " +
product.getElementsByTagName("ar").item(0).getTextContent());

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

   }

}
```

2d) Adatírás

A feladat megoldásához külön metódusokat hoztam létre, ahol az elemek létrehozásra kerülnek, illetve a megfelelő módon összeállítják a szülő-gyerek kapcsolatot.

A gyökérelem annak attribútumai a program elején kerülnek létrehozásra.

Az olyan elemekhez, amelyek tartalmaznak text node-ot (értéket), azokhoz létrehoztam egy külön metódust, ami ezt elvégzi.

A main függvényben létrehoztam minden elemből 5 darabot, majd ezt követően kiírattam fájlba, majd pedig konzolra.

A kód:

```
package hu.domparse.uy5e1l;

import java.io.File;

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

public class DOMWriteUY5E1L {

    public static void main(String[] args) {
        try {
            //DOM létrehozása
            DocumentBuilderFactory dbFactory =
DocumentBuilderFactory.newInstance();
            DocumentBuilder domBuilder = dbFactory.newDocumentBuilder();
            Document doc = domBuilder.newDocument();

            //Gyökér element hozzáadása
            Element root = doc.createElement("webshop");
            root.setAttribute("xmlns:xs", "http://www.w3.org/2001/XMLSchema-
instance");
```

```
            root.setAttribute("xs:noNamespaceSchemaLocation",
"XMLTaskUY5E1L.xsd");
            doc.appendChild(root);

            //Ügyfelek hozzáadása
            root.appendChild(doc.createComment("Ügyfelek"));
            createCustomer(doc, root, "1", "Nagy", "Zsófia", "1122",
"Budapest", "Rózsavölgy ", "7", "+36201234567",
                    "valami@example.com");
            createCustomer(doc, root, "2", "Kovács", "Máté", "2040",
"Budaörs", "Fenyves utca", "12", "+36309876543",
                    "valami@example.com");
            createCustomer(doc, root, "3", "Szabó", "Adrienn", "3300", "Eger",
"Tavasz utca", "21", "+36705551122",
                    "valami@example.com");
            createCustomer(doc, root, "4", "Kiss", "Dániel", "1138",
"Budapest", "Tiszavirág utca", "45", "+36208765432",
                    "valami@example.com");
            createCustomer(doc, root, "5", "Horváth", "Viktória", "9027",
"Győr", "Búzavirág utca", "33", "+36703337788",
                    "valami@example.com");

            //Ügy-Re kapcsolatok hozzáadása
            root.appendChild(doc.createComment("Ügy-Re kapcsolótábla"));
            createCustomerOrderRelationship(doc, root, "1", "2", "1",
"Szeretném, ha SOS ideérne a csomagom!");
            createCustomerOrderRelationship(doc, root, "2", "3", "5",
"Pénteken nem vagyok otthon!");
            createCustomerOrderRelationship(doc, root, "3", "1", "2", "Tegye a
csomagomat a hátsó ajtó elé!");
            createCustomerOrderRelationship(doc, root, "4", "5", "4", "A
csengő nem működik!");
            createCustomerOrderRelationship(doc, root, "5", "4", "3", "A kutya
nem harap! Jöjjön be nyugodtan!");

            //Rendelések hozzáadása
            root.appendChild(doc.createComment("Rendelések"));
            createOrder(doc, root, "1", "aktív", "online", "2", "5", "214.68",
"2023-12-04");
            createOrder(doc, root, "2", "teljesítve", "online", "1", "2",
"2049.98", "2023-12-01");
            createOrder(doc, root, "3", "törölve", "online", "4", "2",
"3649.97", "2023-12-01");
            createOrder(doc, root, "4", "aktív", "online", "5", "3",
"6464.96", "2023-12-04");
            createOrder(doc, root, "5", "fizetésre vár", "online", "3", "1",
"1044.99", "2023-12-03");
```

```
            //Rendelt termékek létrehozása
            root.appendChild(doc.createComment("Rendelt Termékek"));
            createOrderedProduct(doc, root, "1", "4", "3", "1199.99", "27%",
"3");
            createOrderedProduct(doc, root, "2", "2", "5", "999.99", "27%",
"1");
            createOrderedProduct(doc, root, "3", "5", "4", "1599.99", "27%",
"4");
            createOrderedProduct(doc, root, "4", "3", "1", "4.99", "27%",
"32");
            createOrderedProduct(doc, root, "5", "2", "2", "999.99", "27%",
"2");

            //Termékek hozzáadása
            root.appendChild(doc.createComment("Termékek"));
            createProduct(doc, root, "1", "igen", "Logitech MX Master 3S
Egér", "459.99");
            createProduct(doc, root, "2", "igen", "QuantumTech Okosóra",
"999.99");
            createProduct(doc, root, "3", "igen", "Zöld Energiaital Elite
Edition", "4.99");
            createProduct(doc, root, "4", "igen", "Gyorsfőző Pizza Sütő",
"1199.99");
            createProduct(doc, root, "5", "igen", "Kényelmi Zóna
Masszázsfotel", "1599.99");

            //Szállítók hozzáadása
            root.appendChild(doc.createComment("Szállítók"));
            createShipper(doc, root, "1", "ExpressMove Logistics",
"+36201234567", "45", "1037", "Budapest", "Montevideo utca", "14");
            createShipper(doc, root, "2", "RapidCargo Solutions",
"+36309876543", "50", "3529", "Miskolc", "Alkotmány utca", "3");
            createShipper(doc, root, "3", "SwiftShip Express", "+36705551122",
"65", "2045", "Törökbálint", "Tavaszi utca", "21");
            createShipper(doc, root, "4", "PrimeTransit Services",
"+36308765432", "40", "1132", "Budapest", "Váci út", "45");
            createShipper(doc, root, "5", "VelocityFreight", "+36703337788",
"55", "3300", "Eger", "Eperjesi utca", "8");

            //Transformer létrehozása, majd beállítások elvégzése
            TransformerFactory transformerFactory =
TransformerFactory.newInstance();
            Transformer transformer = transformerFactory.newTransformer();
            transformer.setOutputProperty(OutputKeys.ENCODING, "UTF-8");
            transformer.setOutputProperty(OutputKeys.INDENT, "yes");
            transformer.setOutputProperty("{https://xml.apache.org/xslt}indent
-amount", "2");
```

```
            //Kiíratás fájlba
            DOMSource source = new DOMSource(doc);
            File outputFile = new
File("src/hu/domparse/uy5e1l/XMLUY5E1LWrite.xml");
            StreamResult file = new StreamResult(outputFile);
            transformer.transform(source, file);

            //Kiíratás konzolra
            StreamResult console = new StreamResult(System.out);
            transformer.transform(source, console);
        } catch (ParserConfigurationException | TransformerException e) {
             e.printStackTrace();
        }
    }

    public static void createCustomer(Document doc, Element root, String
ugyfelid, String vezeteknev, String keresztnev, String iranyitoszam, String
varos,
            String kozterneve, String hazszam, String telefonszam, String
email) {

        //Elementek létrehozása (ahol szükséges értékkel együtt),
attribútum(ok) csatolása értékkel együtt
        Element customer = doc.createElement("ugyfel");
        customer.setAttribute("ugyfelid", ugyfelid);

        Element name = doc.createElement("nev");
        Element lastName = createElementWithValue(doc, "vezeteknev",
vezeteknev);
        Element firstName = createElementWithValue(doc, "keresztnev",
keresztnev);

        Element address = doc.createElement("lakcim");
        Element postalCode = createElementWithValue(doc, "iranyitoszam",
iranyitoszam);
        Element city = createElementWithValue(doc, "varos", varos);
        Element street = createElementWithValue(doc, "kozterneve",
kozterneve);
        Element houseNumber = createElementWithValue(doc, "hazszam", hazszam);

        Element phoneNumber = createElementWithValue(doc, "telefonszam",
telefonszam);
        Element emailAddress = createElementWithValue(doc, "email", email);

        //Struktúra összeállítása
        name.appendChild(lastName);
        name.appendChild(firstName);
```

```
        address.appendChild(postalCode);
        address.appendChild(city);
        address.appendChild(street);
        address.appendChild(houseNumber);

        customer.appendChild(name);
        customer.appendChild(address);
        customer.appendChild(phoneNumber);
        customer.appendChild(emailAddress);

        //A létrehozott struktúra hozzáadása a gyökérhez
        root.appendChild(customer);
    }

    public static void createCustomerOrderRelationship(Document doc, Element
root, String megjegyzesid, String ugyfelid,
            String rendelesid, String megjegyzes) {

        //Elementek létrehozása (ahol szükséges értékkel együtt),
attribútum(ok) csatolása értékkel együtt
        Element relationship = doc.createElement("ugyre");
        relationship.setAttribute("megjegyzesid", megjegyzesid);
        relationship.setAttribute("ugyfelid", ugyfelid);
        relationship.setAttribute("rendelesid", rendelesid);

        Element comment = createElementWithValue(doc, "megjegyzes",
megjegyzes);

        //Struktúra összeállítása
        relationship.appendChild(comment);

        //A létrehozott struktúra hozzáadása a gyökérhez
        root.appendChild(relationship);
    }

    public static void createOrder(Document doc, Element root, String
rendelesid, String statusz,
            String fizetestipusa, String ugyfelid, String szallitoid, String
osszeg, String datum) {

        //Elementek létrehozása (ahol szükséges értékkel együtt),
attribútum(ok) csatolása értékkel együtt
        Element order = doc.createElement("rendeles");
        order.setAttribute("rendelesid", rendelesid);
        order.setAttribute("statusz", statusz);
        order.setAttribute("fizetestipusa", fizetestipusa);
        order.setAttribute("ugyfelid", ugyfelid);
        order.setAttribute("szallitoid", szallitoid);
```

```
        Element amount = createElementWithValue(doc, "osszeg", osszeg);
        Element orderDate = createElementWithValue(doc, "datum", datum);

        //Struktúra összeállítása
        order.appendChild(amount);
        order.appendChild(orderDate);

        //A létrehozott struktúra hozzáadása a gyökérhez
        root.appendChild(order);
    }

    public static void createOrderedProduct(Document doc, Element root, String
rendelttermekid, String termekid, String rendelesid,
            String egysegar, String afa, String mennyiseg) {

        //Elementek létrehozása (ahol szükséges értékkel együtt),
attribútum(ok) csatolása értékkel együtt
        Element orderedProduct = doc.createElement("rendelttermek");
        orderedProduct.setAttribute("rendelttermekid", rendelttermekid);
        orderedProduct.setAttribute("termekid", termekid);
        orderedProduct.setAttribute("rendelesid", rendelesid);

        Element unitPrice = createElementWithValue(doc, "egysegar", egysegar);
        Element vat = createElementWithValue(doc, "afa", afa);
        Element amount = createElementWithValue(doc, "mennyiseg", mennyiseg);

        //Struktúra összeállítása
        orderedProduct.appendChild(unitPrice);
        orderedProduct.appendChild(vat);
        orderedProduct.appendChild(amount);

        //A létrehozott struktúra hozzáadása a gyökérhez
        root.appendChild(orderedProduct);
    }

    public static void createProduct(Document doc, Element root, String
termekid, String akcios, String nev, String ar) {

        //Elementek létrehozása (ahol szükséges értékkel együtt),
attribútum(ok) csatolása értékkel együtt
        Element product = doc.createElement("termek");
        product.setAttribute("termekid", termekid);
        product.setAttribute("akcios", akcios);

        Element name = createElementWithValue(doc, "nev", nev);
        Element price = createElementWithValue(doc, "ar", ar);
```

```
        //Struktúra összeállítása
        product.appendChild(name);
        product.appendChild(price);

        //A létrehozott struktúra hozzáadása a gyökérhez
        root.appendChild(product);
    }

    public static void createShipper(Document doc, Element root, String
szallitoid, String nev, String telefonszam, String szallitasiar,
            String iranyitoszam, String varos, String kozterneve, String
hazszam) {

        //Elementek létrehozása (ahol szükséges értékkel együtt),
attribútum(ok) csatolása értékkel együtt
        Element shipper = doc.createElement("szallito");
        shipper.setAttribute("szallitoid", szallitoid);

        Element name = createElementWithValue(doc, "nev", nev);
        Element phoneNumber = createElementWithValue(doc, "telefonszam",
telefonszam);
        Element shippingFee = createElementWithValue(doc, "szallitasiar",
szallitasiar);

        Element headQuarters = doc.createElement("kozpontcime");
        Element postalCode = createElementWithValue(doc, "iranyitoszam",
iranyitoszam);
        Element city = createElementWithValue(doc, "varos", varos);
        Element street = createElementWithValue(doc, "kozterneve",
kozterneve);
        Element houseNumber = createElementWithValue(doc, "hazszam", hazszam);

        //Struktúra összeállítása
        headQuarters.appendChild(postalCode);
        headQuarters.appendChild(city);
        headQuarters.appendChild(street);
        headQuarters.appendChild(houseNumber);

        shipper.appendChild(name);
        shipper.appendChild(phoneNumber);
        shipper.appendChild(shippingFee);
        shipper.appendChild(headQuarters);

        //A létrehozott struktúra hozzáadása a gyökérhez
        root.appendChild(shipper);
    }
```

```
    public static Element createElementWithValue(Document doc, String
elementName, String elementValue) {
        //Element létrehozása
        Element elementWithValue = doc.createElement(elementName);
        //Az elementhez hozzáadunk egy text node-ot, ami az értéke lesz
        elementWithValue.appendChild(doc.createTextNode(elementValue));

        //A metódus visszaadja az elementet az értékével együtt
        return elementWithValue;
    }
}
```