

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Rendelések

Készítette: Nyíri Levente

Neptunkód: F023QC

Dátum: 2023.12.01.

Tartalomjegyzék

1. Feladat leírása	3
1.a) Az adatbázis ER modell tervezése	3
1.b) Az ER modell konvertálása XDM modellé	5
1.c) XML dokumentum készítése.....	6
1.d) XML dokumentum alapján XMLSchema készítése	11
2. DOM program készítése az XML dokumentum adatainak adminisztrálása alapján.....	16
2a) adatolvasás.....	16
2.b) adatmódosítás	27
2.c) Adatlekérdezés	43
2.d) Adatírás	52

1. Feladat leírása

A féléves beadandóm témája több étteremhez befutó rendelések nyilvántartása XML-ben. Az alábbi egyedek szerepelnek a nyilvántartásban:

- Futár
- Étterem
- Rendelés
- Termék
- Vevő
- Bankkártya

A fő egyednek az éttermet jelöltem ki, ugyanis ehhez kapcsolódnak a rendelések, valamint a futár egyed. A nyilvántartásomban megtalálhatóak többértékű tulajdonságok, valamint egy összetett tulajdonság, a cím.

Egy vevőnek van bankkártyája és feladhat rendelést egy étteremnek. A rendelésben különböző termékek lehetnek, amelyeknek megadhatjuk a mennyiségét, hogy mennyit szeretnénk belőle rendelni, illetve ha valamelyik összetevőt nem szeretnénk a termékbe, akkor azt is megadhatjuk. Mivel egy termék több rendelésben is szerepelhet és egy rendeléshez több termék is tartozhat, ezért ez N:M kapcsolat lesz, amely az XML dokumentumban külön elemként lesz számontartva. Ebbe az elembe fognak bekerülni a nem kívánt összetevők és a mennyiség is. Az étteremhez tartoznak a futárok. Egy futár csak egy étteremhez tartozhat, de egy étteremhez több futár is. A futároknak van egy tulajdonsága, amely azt adja meg, hogy a futár elérhető-e.

1.a) Az adatbázis ER modell tervezése

Tulajdonságok:

- Futár
 - futárID - elsődleges kulcs
 - név
 - telefonszám
 - elérhető
 - típus – (például autós vagy rolleres)
- Étterem
 - étteremID – elsődleges kulcs
 - cím
 - nyitvatartás
 - elérhetőség
- Rendelés

- rendelésID – elsődleges kulcs
- megjegyzés
- rendelés ideje
- teljes ár

- Termék
 - termékID – elsődleges kulcs
 - név
 - ár
 - elkészítési idő
 - összetevők

- Vevő
 - vevőID – elsődleges kulcs
 - név
 - cím
 - irányítószám
 - közterület neve
 - közterület megnevezése
 - házszám
 - emelet/ajtó
 - telefonszám

- Bankkártya
 - kártyaszám – elsődleges kulcs
 - kártyán szereplő név
 - lejárat dátum
 - biztonsági kód
 - típus

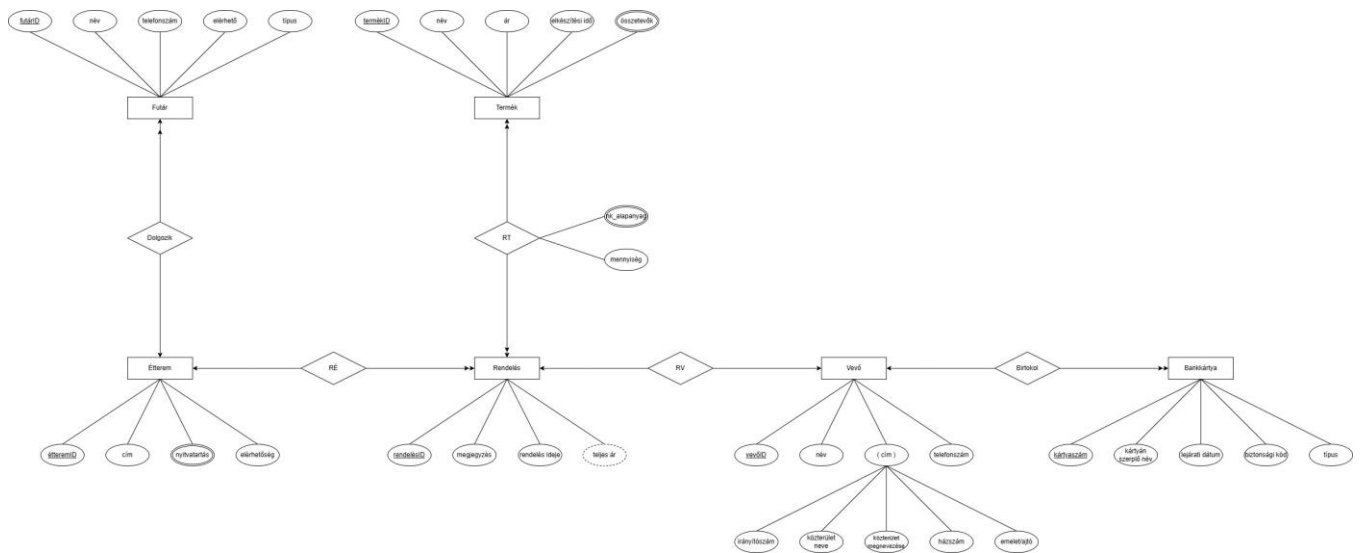
Kapcsolatok:

- Étterem – Futár kapcsolat
 - 1:N kapcsolat, mert egy étteremhez több futár is tartozhat
 - Kapcsolat neve az ER modellen: Dolgozik

- Étterem – Rendelés kapcsolat
 - 1:N kapcsolat, mivel egy étterem akár több rendelést is képes fogadni és feldolgozni

- Kapcsolat neve az ER modellen: RÉ
- Termék – Rendelés kapcsolat
 - N:M kapcsolat
 - Kapcsolat neve az ER modellen: RT
 - Ennek a kapcsolatnak vannak tulajdonságai is:
 - nk_alapanyag – több értékű tulajdonság (nem kívánt alapanyag)
 - mennyiség
- Rendelés – Vevő kapcsolat
 - 1:1 kapcsolat, egy vevőnek egyszerre csak egy rendelése lehet
 - Kapcsolat neve az ER modellen: RV
- Vevő – Bankkártya kapcsolat
 - 1:N kapcsolat, egy vevő akár több bankkártyát is megadhat a rendszerbe
 - Kapcsolat neve az ER modellen: Birtokol

ER Modell:



1.b) Az ER modell konvertálása XDM modellé

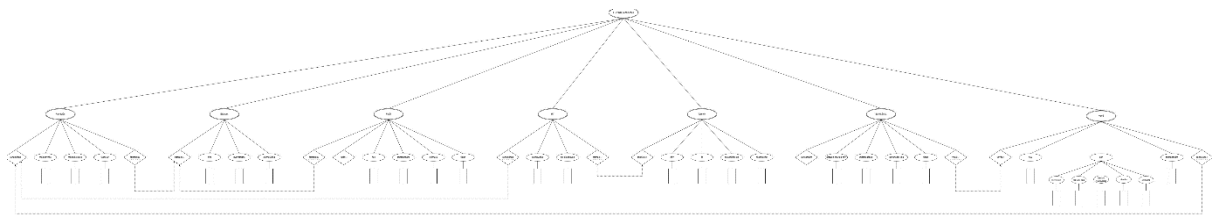
Az ER modell XDM modellé konvertálása során először egy ellipszisben megadtam a gyökérelemet, aminek a neve: F023QC_rendelesek.

Ezután az egyedekből csináltam elemeket, amelyeknek a jelölése szintén egy ellipszis. Az RT (N:M) kapcsolat szintén elem lett.

Az elemekhez megadtam a tulajdonságokat, amelyeket egy kisebb ellipszissel jelöltem. Mindegyikhez tartozik egy téglalap, amely a szöveget jelöli, ez az XML dokumentumban fog megjelenni.

Ezután hozzáadtam a kulcsokat, amelyeket rombuszsal jelöltem. Az elsődleges és az idegen kulcsokat is a rombusz jelöli, viszont az idegen kulcsok szaggatott vonallal vannak aláhúzva.

XDM modell:



1.c) XML dokumentum készítése

Az XDM modell alapján elkészítettem az XML dokumentumot. Először az XML deklarációt írtam be, majd a gyökérelemet, amelyet már az XDM modellben ábrázoltam. Ezután következett, hogy a modellnek megfelelő elemekkel feltöltöttem a gyökérelemet. Ezekbe az elemekbe megadtam a szöveges értékeket.

XML dokumentum forráskódja:

```
<?xml version="1.0" encoding="UTF-8"?>
<F023QC_rendeles xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="XMLSchemaF023QC.xsd">
  <!--Rendelések-->
  <Rendelés rendelésID="1" étteremID="1">
    <megjegyzés>A kutya harap!</megjegyzés>
    <rendelés_ideje>12:00</rendelés_ideje>
    <teljes_ár>15000</teljes_ár>
  </Rendelés>
  <Rendelés rendelésID="2" étteremID="1">
    <rendelés_ideje>14:32</rendelés_ideje>
    <teljes_ár>3000</teljes_ár>
  </Rendelés>
  <Rendelés rendelésID="3" étteremID="2">
    <megjegyzés>Ételalergia</megjegyzés>
    <rendelés_ideje>14:32</rendelés_ideje>
    <teljes_ár>3700</teljes_ár>
  </Rendelés>
</F023QC_rendeles>
```

```

</Rendelés>
<Rendelés rendelésID="4" étteremID="3">
  <megjegyzés>Ételalergia</megjegyzés>
  <rendelés_ideje>14:32</rendelés_ideje>
  <teljes_ár>10000</teljes_ár>
</Rendelés>
<!--Éttermek-->
<Étterem étteremID="1">
  <cím>3528 Miskolc Csokonai Vitéz Mihály utca 40.</cím>
  <nyitvatartás>Hétfő 11:00-19:00</nyitvatartás>
  <nyitvatartás>Kedd 11:00-19:00</nyitvatartás>
  <nyitvatartás>Szerda 11:00-19:00</nyitvatartás>
  <nyitvatartás>Péntek 11:00-19:00</nyitvatartás>
  <elérhetőség>csarda@etterem.hu</elérhetőség>
</Étterem>
<Étterem étteremID="2">
  <cím>3525 Miskolc Szemere Bertalan Utca 3.</cím>
  <nyitvatartás>Hétfő 7:00-18:00</nyitvatartás>
  <nyitvatartás>Kedd 7:00-18:00</nyitvatartás>
  <nyitvatartás>Szerda 7:00-18:00</nyitvatartás>
  <nyitvatartás>Csütörtök 7:00-18:00</nyitvatartás>
  <nyitvatartás>Péntek 7:00-18:00</nyitvatartás>
  <nyitvatartás>Szombat 8:00-18:00</nyitvatartás>
  <nyitvatartás>Vasárnap 10:00-15:00</nyitvatartás>
  <elérhetőség>https://falankfanny.hu/</elérhetőség>
</Étterem>
<Étterem étteremID="3">
  <cím>3526 Miskolc, Petőfi Tér 1.</cím>
  <nyitvatartás>Hétfő 8:00-23:00</nyitvatartás>
  <nyitvatartás>Kedd 8:00-23:00</nyitvatartás>
  <nyitvatartás>Szerda 8:00-23:00</nyitvatartás>
  <nyitvatartás>Csütörtök 8:00-23:00</nyitvatartás>
  <nyitvatartás>Péntek 8:00-23:00</nyitvatartás>
  <elérhetőség>http://www.barkafood.hu/</elérhetőség>
</Étterem>
<!--Futárok-->
<Futár futárID="1" étteremID="1">
  <név>Nagy Milán</név>
  <telefonszám>0630 320 4870</telefonszám>
  <elérhető>0</elérhető>
  <típus>Biciklis</típus>
</Futár>
<Futár futárID="2" étteremID="1">
  <név>Kovács Aladár</név>
  <telefonszám>0630 560 8893</telefonszám>
  <elérhető>0</elérhető>
  <típus>Autós</típus>
</Futár>
<Futár futárID="3" étteremID="1">
  <név>Kiss Béla</név>
  <telefonszám>0630 632 4856</telefonszám>
  <elérhető>1</elérhető>
  <típus>Autós</típus>
</Futár>
<Futár futárID="4" étteremID="2">
  <név>Varga Roland</név>
  <telefonszám>0620 569 4236</telefonszám>
  <elérhető>1</elérhető>
  <típus>Rolleres</típus>
</Futár>

```

```

<Futár futárID="5" étteremID="2">
  <név>Kovács Péter</név>
  <telefonszám>0646 329 4568</telefonszám>
  <elérhető>0</elérhető>
  <típus>Biciklis</típus>
</Futár>
<Futár futárID="6" étteremID="3">
  <név>Szabó Éva</név>
  <telefonszám>0620 563 7821</telefonszám>
  <elérhető>1</elérhető>
  <típus>Autós</típus>
</Futár>
<!--Rendelés - Termék kapcsolat-->
<RT rendelésID="1" termékID="1">
  <mennyiség>1</mennyiség>
  <nk_összetevő>Répa</nk_összetevő>
</RT>
<RT rendelésID="1" termékID="3">
  <mennyiség>4</mennyiség>
</RT>
<RT rendelésID="1" termékID="5">
  <mennyiség>1</mennyiség>
  <nk_összetevő>Porcukor</nk_összetevő>
</RT>
<RT rendelésID="2" termékID="1">
  <mennyiség>1</mennyiség>
  <nk_összetevő>Zeller</nk_összetevő>
</RT>
<RT rendelésID="2" termékID="5">
  <mennyiség>1</mennyiség>
  <nk_összetevő>Porcukor</nk_összetevő>
</RT>
<RT rendelésID="3" termékID="2">
  <mennyiség>5</mennyiség>
  <nk_összetevő>Liszt</nk_összetevő>
  <nk_összetevő>Vanília Kivonat</nk_összetevő>
</RT>
<RT rendelésID="3" termékID="4">
  <mennyiség>4</mennyiség>
</RT>
<RT rendelésID="4" termékID="5">
  <mennyiség>5</mennyiség>
  <nk_összetevő>Só</nk_összetevő>
  <nk_összetevő>Porcukor</nk_összetevő>
  <nk_összetevő>Tejföl</nk_összetevő>
</RT>
<!--Termékek-->
<Termék termékID="1">
  <név>Húsleves</név>
  <ár>1000</ár>
  <elkészítési_idő>30 perc</elkészítési_idő>
  <összetevők>sárgarépa</összetevők>
  <összetevők>fehérrépa</összetevők>
  <összetevők>karalábé</összetevők>
  <összetevők>zeller</összetevők>
  <összetevők>vöröshagyma</összetevők>
  <összetevők>paprika</összetevők>
  <összetevők>petrezselyem</összetevők>
  <összetevők>só</összetevők>
  <összetevők>bors</összetevők>

```



```
<összetevők>tészta</összetevők>
</Termék>
<Termék termékID="2">
  <név>Brownie</név>
  <ár>500</ár>
  <elkészítési_idő>15 perc</elkészítési_idő>
  <összetevők>liszt</összetevők>
  <összetevők>só</összetevők>
  <összetevők>kakaópor</összetevők>
  <összetevők>étcsokoládé</összetevők>
  <összetevők>vaj</összetevők>
  <összetevők>nutella</összetevők>
  <összetevők>tojás</összetevők>
  <összetevők>cukor</összetevők>
  <összetevők>vanília kivonat</összetevők>
</Termék>
<Termék termékID="3">
  <név>Omlós libacomb</név>
  <ár>4000</ár>
  <elkészítési_idő>130 perc</elkészítési_idő>
  <összetevők>libacomb</összetevők>
  <összetevők>tengeri só</összetevők>
</Termék>
<Termék termékID="4">
  <név>Muffin</név>
  <ár>300</ár>
  <elkészítési_idő>30 perc</elkészítési_idő>
  <összetevők>tojás</összetevők>
  <összetevők>cukor</összetevők>
  <összetevők>vaj</összetevők>
  <összetevők>tej</összetevők>
  <összetevők>liszt</összetevők>
  <összetevők>só</összetevők>
  <összetevők>sütőpor</összetevők>
  <összetevők>vaníliás cukor</összetevők>
</Termék>
<Termék termékID="5">
  <név>Túrógombóc</név>
  <ár>2000</ár>
  <elkészítési_idő>25 perc</elkészítési_idő>
  <összetevők>túró</összetevők>
  <összetevők>tojás</összetevők>
  <összetevők>búzadara</összetevők>
  <összetevők>só</összetevők>
  <összetevők>szódabikarbóna</összetevők>
  <összetevők>zsemlemorzsa</összetevők>
  <összetevők>olaj</összetevők>
  <összetevők>porcukor</összetevők>
  <összetevők>tejföl</összetevők>
</Termék>
<!--Bankkártyák-->
<Bankkártya kártyaszám="1174859659985563" vevőID="1">
  <kártyán_szereplő_név>Nyíri Levente</kártyán_szereplő_név>
  <lejárat_i_dátum>12/26</lejárat_i_dátum>
  <biztonsági_kód>056</biztonsági_kód>
  <típus>Mastercard</típus>
</Bankkártya>
<Bankkártya kártyaszám="1174859655585888" vevőID="1">
  <kártyán_szereplő_név>Nyíri Levente</kártyán_szereplő_név>
  <lejárat_i_dátum>05/28</lejárat_i_dátum>
```

```
<biztonsági_kód>356</biztonsági_kód>
<típus>Visa</típus>
</Bankkártya>
<Bankkártya kártyaszám="1174865851185111" vevőID="1">
  <kártyán_szureplő_név>Nyíri Levente</kártyán_szureplő_név>
  <lejáratí_dátum>12/23</lejáratí_dátum>
  <biztonsági_kód>111</biztonsági_kód>
  <típus>Mastercard</típus>
</Bankkártya>
<Bankkártya kártyaszám="1176834769216856" vevőID="2">
  <kártyán_szureplő_név>Nagyné Kovács Rebeka</kártyán_szureplő_név>
  <lejáratí_dátum>12/23</lejáratí_dátum>
  <biztonsági_kód>111</biztonsági_kód>
  <típus>Visa</típus>
</Bankkártya>
<Bankkártya kártyaszám="1179863452169933" vevőID="2">
  <kártyán_szureplő_név>Nagy Balázs</kártyán_szureplő_név>
  <lejáratí_dátum>02/27</lejáratí_dátum>
  <biztonsági_kód>986</biztonsági_kód>
  <típus>Visa</típus>
</Bankkártya>
<Bankkártya kártyaszám="1175453336278779" vevőID="3">
  <kártyán_szureplő_név>Kis István</kártyán_szureplő_név>
  <lejáratí_dátum>01/29</lejáratí_dátum>
  <biztonsági_kód>494</biztonsági_kód>
  <típus>Mastercard</típus>
</Bankkártya>
<Bankkártya kártyaszám="1171154896532478" vevőID="4">
  <kártyán_szureplő_név>Varga Iván</kártyán_szureplő_név>
  <lejáratí_dátum>01/24</lejáratí_dátum>
  <biztonsági_kód>555</biztonsági_kód>
  <típus>Visa</típus>
</Bankkártya>
<!--Vevők-->
<Vevő rendelésID="1" vevőID="1">
  <név>Nyíri Levente</név>
  <telefonszám>0620 589 9938</telefonszám>
  <cím>
    <irányítószám>3528</irányítószám>
    <közterület_neve>Csokonai Vitéz Mihály</közterület_neve>
    <közterület_megnevezése>utca</közterület_megnevezése>
    <házszám>11</házszám>
  </cím>
</Vevő>
<Vevő rendelésID="2" vevőID="2">
  <név>Nagy Balázs</név>
  <telefonszám>0630 597 6413</telefonszám>
  <cím>
    <irányítószám>3535</irányítószám>
    <közterület_neve>Kuruc</közterület_neve>
    <közterület_megnevezése>utca</közterület_megnevezése>
    <házszám>23</házszám>
    <emelet_ajtó>5/1</emelet_ajtó>
  </cím>
</Vevő>
<Vevő rendelésID="3" vevőID="3">
  <név>Kis István</név>
  <telefonszám>0620 365 5876</telefonszám>
  <cím>
    <irányítószám>3534</irányítószám>
```

```

        <közterület_neve>Nagy Lajos király</közterület_neve>
        <közterület_megnevezése>út</közterület_megnevezése>
        <házzsám>30</házzsám>
    </cím>
</Vevő>
<Vevő rendelésID="4" vevőID="4">
    <név>Varga Iván</név>
    <telefonszám>0630 879 1298</telefonszám>
    <cím>
        <irányítószám>3531</irányítószám>
        <közterület_neve>Aba</közterület_neve>
        <közterület_megnevezése>utca</közterület_megnevezése>
        <házzsám>2</házzsám>
        <emelet_ajtó>3/4</emelet_ajtó>
    </cím>
</Vevő>
</F023QC_rendeletek>

```

1.d) XML dokumentum alapján XMLSchema készítése

Először kigyűjtöttem az egyszerű típusokat, majd meghatároztam a saját típusokat.

Ezek a következők:

- Telefonszám típus
 - Először 4 karakter majd 3 és végül 3 karakter szóközzel elválasztva
- Igaz/Hamis típus
 - 0 vagy 1 értéket vehet fel
- Irányítószám típus
 - maximum 4 számjegyű és int típusú
- Kártyaszám típus
 - Beállítottam, hogy hány számjegyű és azt, hogy long típusú
- Emelet/ajtó típus
 - Lennie kell egy / jelnek benne
- Közterület megnevezése
 - Megadtam, hogy milyen értékeket vehet fel
- Biztonsági kód
 - Megadtam, hogy maximum 3 karakter hosszú kell, hogy legyen
- Lejárat dátum
 - rendelkezzen egy / jellel és mindkét oldalán 2 hosszúságú karakter kell, hogy legyen

Ezután megadtam az összetett típusokat, majd a gyökérelemtől kezdve felépítettem formátumot. Elsődleges és idegen kulcsokat is létrehoztam, illetve az 1:1 kapcsolatot is definiáltam.

XMLSchema forráskódja:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

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

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

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

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

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

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

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

  <xs:element name="telefonszám" type="telefonszámType"/>

  <xs:element name="elérhető" type="boolType"/>

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

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

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

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

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

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

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

  <xs:element name="lejárat_i dátum" type="lejárat_i dátumType"/>

  <xs:element name="biztonsági_kód" type="biztonsági_kódType"/>

  <xs:element name="irányítószám" type="irányítószámType"/>

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

  <xs:element name="közterület_megnevezése" type="közterület_megnevezéseType"/>

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

  <xs:element name="emelet_ajtó" type="emelet_ajtóType"/>

  <xs:simpleType name="telefonszámType">
    <xs:restriction base="xs:string">
      <xs:pattern value="\d{4} \d{3} \d{4}" />
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

```

</xs:simpleType>

<xs:simpleType name="boolType">
  <xs:restriction base="xs:boolean" />
</xs:simpleType>

<xs:simpleType name="irányítószámType">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="1000"/>
    <!-- Csak 4 számjegyű lehet az irányítószám -->
    <xs:maxInclusive value="9999"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="kártyaszámType">
  <xs:restriction base="xs:long">
    <xs:minInclusive value="10000000000000000"/>
    <!-- Kártyaszám hossza -->
    <xs:maxInclusive value="9999999999999999"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="emelet_ajtóType">
  <xs:restriction base="xs:string">
    <!-- Legyen benne egy / jel -->
    <xs:pattern value=".*\/.*"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="közterület_megnevezéseType">
  <xs:restriction base="xs:string">
    <!-- utca/út -->
    <xs:enumeration value="utca"/>
    <xs:enumeration value="út"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="biztonsági_kódType">
  <xs:restriction base="xs:string">
    <xs:length value="3" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="lejárat_i dátumType">
  <xs:restriction base="xs:string">
    <!-- Legyen benne egy / jel -->
    <xs:pattern value="\d{2}\/\d{2}"/>
  </xs:restriction>
</xs:simpleType>

<!-- Összetett típusok -->

<xs:complexType name = "RendelésType">
  <xs:sequence>
    <xs:element ref="megjegyzés" minOccurs="0" maxOccurs="1"/>
    <xs:element ref="rendelés_ideje" />
    <xs:element ref="teljes ár" />
  </xs:sequence>

```

```

        <xs:attribute name="rendelésID" type="xs:integer" use="required" />
        <xs:attribute name="étteremID" type="xs:integer" use="required" />
    </xs:complexType>

    <xs:complexType name="ÉtteremType">
        <xs:sequence>
            <xs:element ref="cím" />
            <xs:element ref="nyitvatartás" minOccurs="1" maxOccurs="unbounded"/>
            <xs:element ref="elérhetőség" />
        </xs:sequence>
        <xs:attribute name="étteremID" type="xs:integer" use="required" />
    </xs:complexType>

    <xs:complexType name="FutárType">
        <xs:sequence>
            <xs:element ref="név" />
            <xs:element ref="telefonszám" />
            <xs:element ref="elérhető" />
            <xs:element ref="típus" />
        </xs:sequence>
        <xs:attribute name="futárID" type="xs:integer" use="required" />
        <xs:attribute name="étteremID" type="xs:integer" use="required" />
    </xs:complexType>

    <xs:complexType name="RTType">
        <xs:sequence>
            <xs:element ref="mennyiség" />
            <xs:element ref="nkösszetevő" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="rendelésID" type="xs:integer" use="required" />
        <xs:attribute name="termékID" type="xs:integer" use="required" />
    </xs:complexType>

    <xs:complexType name="TermékType">
        <xs:sequence>
            <xs:element ref="név" />
            <xs:element ref="ár" />
            <xs:element ref="elkészítési_idő" />
            <xs:element ref="összetevők" minOccurs="1" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="termékID" type="xs:integer" use="required" />
    </xs:complexType>

    <xs:complexType name="BankkártyaType">
        <xs:sequence>
            <xs:element ref="kártyán_szereplő_név" />
            <xs:element ref="lejáratí_dátum" />
            <xs:element ref="biztonsági_kód" />
            <xs:element ref="típus" />
        </xs:sequence>
        <xs:attribute name="kártyaszám" type="kártyaszámType" use="required" />
        <xs:attribute name="vevőID" type="xs:integer" use="required" />
    </xs:complexType>

    <xs:complexType name="VevőType">
        <xs:sequence>
            <xs:element ref="név" />
            <xs:element ref="telefonszám" />
            <xs:element name="cím">
                <xs:complexType>

```

```

        <xs:sequence>
            <xs:element ref="irányítószám" />
            <xs:element ref="közterület_neve" />
            <xs:element ref="közterület_megnevezése" />
            <xs:element ref="házszám" />
            <xs:element ref="emelet_ajtó" minOccurs="0" maxOccurs="1"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="vevőID" type="xs:integer" use="required" />
<xs:attribute name="rendelésID" type="xs:integer" use="required" />
</xs:complexType>

<!-- Felépítés a gyökérelemtől -->
<xs:element name="F023QC_rendeletek">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Rendelés" type="RendelésType" minOccurs="1"
maxOccurs="unbounded"/>
            <xs:element name="Étterem" type="ÉtteremType" minOccurs="1"
maxOccurs="unbounded"/>
            <xs:element name="Futár" type="FutárType" minOccurs="1"
maxOccurs="unbounded"/>
            <xs:element name="RT" type="RTType" minOccurs="1"
maxOccurs="unbounded"/>
            <xs:element name="Termék" type="TermékType" minOccurs="1"
maxOccurs="unbounded"/>
            <xs:element name="Bankkártya" type="BankkártyaType" minOccurs="1"
maxOccurs="unbounded"/>
            <xs:element name="Vevő" type="VevőType" minOccurs="1"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>

    <!-- Kulcsok -->
    <xs:key name="rendelésPK">
        <xs:selector xpath="Rendelés"/>
        <xs:field xpath="@rendelésID"/>
    </xs:key>

    <xs:key name="étteremPK">
        <xs:selector xpath="Étterem"/>
        <xs:field xpath="@étteremID"/>
    </xs:key>

    <xs:key name="futárPK">
        <xs:selector xpath="Futár"/>
        <xs:field xpath="@futárID"/>
    </xs:key>

    <xs:key name="termékPK">
        <xs:selector xpath="Termék"/>
        <xs:field xpath="@termékID"/>
    </xs:key>

    <xs:key name="kártyaszámPK">
        <xs:selector xpath="Bankkártya"/>
        <xs:field xpath="@kártyaszám"/>
    </xs:key>

    <xs:key name="vevőPK">
        <xs:selector xpath="Vevő"/>
        <xs:field xpath="@vevőID"/>
    </xs:key>

    <!-- Idegen kulcsok -->

```

```

<xs:keyref name="Rendelés_RT_FK" refer="rendelésPK">
  <xs:selector xpath="RT" />
  <xs:field xpath="@rendelésID" />
</xs:keyref>

<xs:keyref name="Rendelés_Vevő_FK" refer="rendelésPK">
  <xs:selector xpath="Vevő" />
  <xs:field xpath="@rendelésID" />
</xs:keyref>

<xs:keyref name="Étterem_Rendelés_FK" refer="étteremPK">
  <xs:selector xpath="Rendelés" />
  <xs:field xpath="@étteremID" />
</xs:keyref>

<xs:keyref name="Étterem_Futár_FK" refer="étteremPK">
  <xs:selector xpath="Futár" />
  <xs:field xpath="@étteremID" />
</xs:keyref>

<xs:keyref name="Termék_RT_FK" refer="termékPK">
  <xs:selector xpath="RT" />
  <xs:field xpath="@termékID" />
</xs:keyref>

<xs:keyref name="Vevő_Bakkártya_FK" refer="vevőPK">
  <xs:selector xpath="Bankkártya" />
  <xs:field xpath="@vevőID" />
</xs:keyref>

<!-- 1:1 kapcsolat -->
<xs:unique name="VevoRendeles">
  <xs:selector xpath="Vevő"/>
  <xs:field xpath="@rendelésID"/>
</xs:unique>

</xs:element>

</xs:schema>

```

2. DOM program készítése az XML dokumentum adatainak adminisztrálása alapján

2a) adatolvasás

Ebben a feladatban egy olyan java osztályt hoztam létre, amely beolvas egy xml dokumentumot, majd annak tartalmát kiírja a consolra és egy fájlba. Az osztályban az alábbi metódusokat hoztam létre:

- removeText
 - Ezzel a függvénnyel a dokumentumban lévő üres text nodeokat törlöm ki
- printDocument

- A függvény célja, hogy a dokumentumot a gyökérelemtől kezdve strukturálisan kiírja a console-ra

- writeXML
 - Ezzel a függvénnyel írom ki egy új fájlba a dokumentum tartalmát. Itt használom fel a removeText metódust
- getIndent
 - Célja, hogy a kiíratás során visszaadjon az elem szintjének megfelelő behúzást, ezzel olvashatóbbá téve a kiíratást.

Forráskód:

```
package hu.domparse.F023QC;

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

public class DOMReadF023QC {
    public static void main(String[] args) {
        try {
            File xmlFile = new File("XMLF023QC.xml");
            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
            DocumentBuilder dBuilder = factory.newDocumentBuilder();
            Document document = dBuilder.parse(xmlFile);
            document.getDocumentElement().normalize();

            //Kiíratom a fájlt
            printDocument(document);
        }
    }
}
```

```

//Új fájl létrehozása

writeXML(document);

}catch(Exception e){

System.out.println(e.getMessage());

}}

//Törölöm a felesleges text nodeokat

private static void removeText(Node root) {

//Kigyűjtöm a routnak a gyerekeit, majd végigmegyek rajta

NodeList nodeList = root.getChildNodes();

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

//Ha találok felesleget, akkor kitörölöm azt

if (nodeList.item(i).getNodeType() == Node.TEXT_NODE &&
nodeList.item(i).getTextContent().strip().isEmpty()) {

nodeList.item(i).getParentNode().removeChild(nodeList.item(i));

//A törlés miatt 1-el előrébb járunk, mint ahol kéne, ezért az i-t csökkentem egyel

i--;

//Ha nem felesleges node, akkor ezt a node-ot is leellenőrzöm és kitörölöm belőle a felesleget

} else {

removeText(nodeList.item(i));

}}}

private static void printDocument(Document doc) {

//Kiíratom az xml deklarációt

System.out.println("<?xml version=\""+doc.getXmlVersion()+"\"
encoding=\""+doc.getXmlEncoding()+"\"?>");

//Kiíratom a gyökérelemet

Element rootelement =doc.getDocumentElement();

//Első sora a gyökérelemnek

String rootRow ="<"+rootelement.getNodeName()+" ";

```

```

//Gyökérelem "attribútumainak" hozzáadása az első sorhoz

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

if(i==rootelement.getAttributes().getLength()-1) {

rootRow +=
rootelement.getAttributes().item(i).getNodeName()+"=\""+rootelement.getAttributes().item(i).getTextContent()
+"\">";

}else {

rootRow +=
rootelement.getAttributes().item(i).getNodeName()+"=\""+rootelement.getAttributes().item(i).getTextContent()
+"\" ";

}}

//Első sor kiírása

System.out.println(rootRow);

//A gyökérelemben lévő összes elem kiírása

NodeList elements = rootelement.getChildNodes();

//Végigmegyek az elemeken

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

Node currentElement = elements.item(i);

//Ha a jelenlegi elem komment, akkor kiíratom kommentként

if(currentElement.getNodeType()==Node.COMMENT_NODE) {

System.out.println(getIndent(1)+"<!--"+currentElement.getTextContent()+"-->");

//Ellenkező esetben kiíratom normál Node-ként

}else if (currentElement.getNodeType()==Node.ELEMENT_NODE) {

//Az első sora az elemnek

String fRow ="<"+currentElement.getNodeName()+" ";

//Hozzáadom az attribútumokat az első sorhoz

for(int j = 0; j< currentElement.getAttributes().getLength();j++) {

if(j==currentElement.getAttributes().getLength()-1) {

```

```

fRow +=
currentElement.getAttributes().item(j).getNodeName()+"=\""+currentElement.getAttributes().item(j).getTextCo
ntent()+"\">";

}else {

fRow +=
currentElement.getAttributes().item(j).getNodeName()+"=\""+currentElement.getAttributes().item(j).getTextCo
ntent()+"\" ";

}}

//Kiíratom az első sort

System.out.println(getIndent(1)+fRow);

//Kigyűjtöm az elem gyerekelemeit

NodeList childs = currentElement.getChildNodes();

for(int j=0;j<childs.getLength();j++) {

Node child = childs.item(j);

//Ha a gyerekelem típusa ELEMENT

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

//Ha csak sima gyerekelem, akkor kiíratom (Csak szöveg van benne)

if(child.getChildNodes().getLength()==1) {

System.out.println(getIndent(2)+"<"+child.getNodeName()+">"+child.getTextContent()+"</"+child.getNodeNa
me()+">");

//Ha nem csak szöveg van benne, akkor meg kell nézni a gyerekelem gyerekelemeit és azokat is kiíratni (cím
esetében)

}else {

//Kiírom a gyerekelemet

System.out.println(getIndent(2)+"<"+child.getNodeName()+">");

//Kiírom a gyerekelem gyerekelemeit

for(int k =0;k<child.getChildNodes().getLength();k++) {

if(child.getChildNodes().item(k).getNodeType()==Node.ELEMENT_NODE) {

System.out.println(getIndent(3)+"<"+child.getChildNodes().item(k).getNodeName()+">"+child.getChildNodes(
).item(k).getTextContent()+"</"+child.getChildNodes().item(k).getNodeName()+">");

}}

//Bezárom a gyerekelemet

System.out.println(getIndent(2)+"</"+child.getNodeName()+">");

}}}

```

```

//Bezárom az elemet

System.out.println(getIndent(1)+"</"+currentElement.getNodeName()+">");

}}

//Bezárom a gyökérelemet

System.out.println("</"+rootelement.getNodeName()+">");

}

private static void writeXML(Document doc) {

try {

//</text> node-októl megtisztítom a dokumentumot, majd a transformer segítségével kiíratom egy fájlba

removeText(doc);

TransformerFactory transformerFactory = TransformerFactory.newInstance();

Transformer transformer = transformerFactory.newTransformer();

transformer.setOutputProperty(OutputKeys.ENCODING, "UTF-8");

transformer.setOutputProperty(OutputKeys.INDENT, "yes");

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

FileWriter fileWriter = new FileWriter("XMLF023QC1.xml");

transformer.transform(new DOMSource(doc), new StreamResult(fileWriter));

System.out.println("Sikeres kiírás fájlba!");

} catch (Exception e) {

e.printStackTrace();

}}

//Behúzások meghatározása

private static String getIndent(int indent) {

//A paraméternek megfelelő behúzást ad vissza, a behúzás 4 darab space

String ret="";

for(int i=0;i<indent;i++) {

ret+=" ";

}

return ret;
}

```

Output:

```

<?xml version="1.0" encoding="UTF-8"?>
<F023QC_rendelesek xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="XMLSchemaF023QC.xsd">

```

```

<!--Rendelések-->
<Rendelés rendelésID="1" étteremID="1">
    <megjegyzés>A kutya harap!</megjegyzés>
    <rendelés_ideje>12:00</rendelés_ideje>
    <teljes_ár>15000</teljes_ár>
</Rendelés>
<Rendelés rendelésID="2" étteremID="1">
    <rendelés_ideje>14:32</rendelés_ideje>
    <teljes_ár>3000</teljes_ár>
</Rendelés>
<Rendelés rendelésID="3" étteremID="2">
    <megjegyzés>Ételalergia</megjegyzés>
    <rendelés_ideje>14:32</rendelés_ideje>
    <teljes_ár>3700</teljes_ár>
</Rendelés>
<Rendelés rendelésID="4" étteremID="3">
    <megjegyzés>Ételalergia</megjegyzés>
    <rendelés_ideje>14:32</rendelés_ideje>
    <teljes_ár>10000</teljes_ár>
</Rendelés>
<!--Éttermek-->
<Étterem étteremID="1">
    <cím>3528 Miskolc Csokonai Vitéz Mihály utca 40.</cím>
    <nyitvatartás>Hétfő 11:00-19:00</nyitvatartás>
    <nyitvatartás>Kedd 11:00-19:00</nyitvatartás>
    <nyitvatartás>Szerda 11:00-19:00</nyitvatartás>
    <nyitvatartás>Péntek 11:00-19:00</nyitvatartás>
    <elérhetőség>csarda@etterem.hu</elérhetőség>
</Étterem>
<Étterem étteremID="2">
    <cím>3525 Miskolc Szemere Bertalan Utca 3.</cím>
    <nyitvatartás>Hétfő 7:00-18:00</nyitvatartás>
    <nyitvatartás>Kedd 7:00-18:00</nyitvatartás>
    <nyitvatartás>Szerda 7:00-18:00</nyitvatartás>
    <nyitvatartás>Csütörtök 7:00-18:00</nyitvatartás>
    <nyitvatartás>Péntek 7:00-18:00</nyitvatartás>
    <nyitvatartás>Szombat 8:00-18:00</nyitvatartás>
    <nyitvatartás>Vasárnap 10:00-15:00</nyitvatartás>
    <elérhetőség>https://falankfanny.hu/</elérhetőség>
</Étterem>
<Étterem étteremID="3">
    <cím>3526 Miskolc, Petőfi Tér 1.</cím>
    <nyitvatartás>Hétfő 8:00-23:00</nyitvatartás>
    <nyitvatartás>Kedd 8:00-23:00</nyitvatartás>
    <nyitvatartás>Szerda 8:00-23:00</nyitvatartás>
    <nyitvatartás>Csütörtök 8:00-23:00</nyitvatartás>
    <nyitvatartás>Péntek 8:00-23:00</nyitvatartás>
    <elérhetőség>http://www.barkafood.hu/</elérhetőség>
</Étterem>
<!--Futárok-->
<Futár futárID="1" étteremID="1">
    <név>Nagy Milán</név>
    <telefonszám>0630 320 4870</telefonszám>
    <elérhető>0</elérhető>
    <típus>Biciklis</típus>
</Futár>
<Futár futárID="2" étteremID="1">
    <név>Kovács Aladár</név>
    <telefonszám>0630 560 8893</telefonszám>
    <elérhető>0</elérhető>

```

```

        <típus>Autós</típus>
</Futár>
<Futár futárID="3" étteremID="1">
    <név>Kiss Béla</név>
    <telefonszám>0630 632 4856</telefonszám>
    <elérhető>1</elérhető>
    <típus>Autós</típus>
</Futár>
<Futár futárID="4" étteremID="2">
    <név>Varga Roland</név>
    <telefonszám>0620 569 4236</telefonszám>
    <elérhető>1</elérhető>
    <típus>Rolleres</típus>
</Futár>
<Futár futárID="5" étteremID="2">
    <név>Kovács Péter</név>
    <telefonszám>0646 329 4568</telefonszám>
    <elérhető>0</elérhető>
    <típus>Biciklis</típus>
</Futár>
<Futár futárID="6" étteremID="3">
    <név>Szabó Éva</név>
    <telefonszám>0620 563 7821</telefonszám>
    <elérhető>1</elérhető>
    <típus>Autós</típus>
</Futár>
<!--Rendelés - Termék kapcsolat-->
<RT rendelésID="1" termékID="1">
    <mennyiség>1</mennyiség>
    <nk_összetevő>Répa</nk_összetevő>
</RT>
<RT rendelésID="1" termékID="3">
    <mennyiség>4</mennyiség>
</RT>
<RT rendelésID="1" termékID="5">
    <mennyiség>1</mennyiség>
    <nk_összetevő>Porcukor</nk_összetevő>
</RT>
<RT rendelésID="2" termékID="1">
    <mennyiség>1</mennyiség>
    <nk_összetevő>Zeller</nk_összetevő>
</RT>
<RT rendelésID="2" termékID="5">
    <mennyiség>1</mennyiség>
    <nk_összetevő>Porcukor</nk_összetevő>
</RT>
<RT rendelésID="3" termékID="2">
    <mennyiség>5</mennyiség>
    <nk_összetevő>Liszt</nk_összetevő>
    <nk_összetevő>Vanília Kivonat</nk_összetevő>
</RT>
<RT rendelésID="3" termékID="4">
    <mennyiség>4</mennyiség>
</RT>
<RT rendelésID="4" termékID="5">
    <mennyiség>5</mennyiség>
    <nk_összetevő>Só</nk_összetevő>
    <nk_összetevő>Porcukor</nk_összetevő>
    <nk_összetevő>Tejföl</nk_összetevő>
</RT>

```



```
<!--Termékek-->
<Termék termékID="1">
  <név>Húsleves</név>
  <ár>1000</ár>
  <elkészítési_idő>30 perc</elkészítési_idő>
  <összetevők>sárgarépa</összetevők>
  <összetevők>fehérrépa</összetevők>
  <összetevők>karalábé</összetevők>
  <összetevők>zeller</összetevők>
  <összetevők>vöröshagyma</összetevők>
  <összetevők>paprika</összetevők>
  <összetevők>petrezselyem</összetevők>
  <összetevők>só</összetevők>
  <összetevők>bors</összetevők>
  <összetevők>tészta</összetevők>
</Termék>
<Termék termékID="2">
  <név>Brownie</név>
  <ár>500</ár>
  <elkészítési_idő>15 perc</elkészítési_idő>
  <összetevők>liszt</összetevők>
  <összetevők>só</összetevők>
  <összetevők>kakaópor</összetevők>
  <összetevők>étcsokoládé</összetevők>
  <összetevők>vaj</összetevők>
  <összetevők>nutella</összetevők>
  <összetevők>tojás</összetevők>
  <összetevők>cukor</összetevők>
  <összetevők>vanília kivonat</összetevők>
</Termék>
<Termék termékID="3">
  <név>Omlós libacomb</név>
  <ár>4000</ár>
  <elkészítési_idő>130 perc</elkészítési_idő>
  <összetevők>libacomb</összetevők>
  <összetevők>tengeri só</összetevők>
</Termék>
<Termék termékID="4">
  <név>Muffin</név>
  <ár>300</ár>
  <elkészítési_idő>30 perc</elkészítési_idő>
  <összetevők>tojás</összetevők>
  <összetevők>cukor</összetevők>
  <összetevők>vaj</összetevők>
  <összetevők>tej</összetevők>
  <összetevők>liszt</összetevők>
  <összetevők>só</összetevők>
  <összetevők>sütőpor</összetevők>
  <összetevők>vaníliás cukor</összetevők>
</Termék>
<Termék termékID="5">
  <név>Túrógombóc</név>
  <ár>2000</ár>
  <elkészítési_idő>25 perc</elkészítési_idő>
  <összetevők>túró</összetevők>
  <összetevők>tojás</összetevők>
  <összetevők>búzadara</összetevők>
  <összetevők>só</összetevők>
  <összetevők>szódabikarbóna</összetevők>
  <összetevők>zsemlemorzsa</összetevők>
```

```

        <összetevők>olaj</összetevők>
        <összetevők>porcukor</összetevők>
        <összetevők>tejföl</összetevők>
    </Termék>
    <!--Bankkártyák-->
    <Bankkártya kártyaszám="1174859659985563" vevőID="1">
        <kártyán_szereplő_név>Nyíri Levente</kártyán_szereplő_név>
        <lejárat_i_dátum>12/26</lejárat_i_dátum>
        <biztonsági_kód>056</biztonsági_kód>
        <típus>Mastercard</típus>
    </Bankkártya>
    <Bankkártya kártyaszám="1174859655585888" vevőID="1">
        <kártyán_szereplő_név>Nyíri Levente</kártyán_szereplő_név>
        <lejárat_i_dátum>05/28</lejárat_i_dátum>
        <biztonsági_kód>356</biztonsági_kód>
        <típus>Visa</típus>
    </Bankkártya>
    <Bankkártya kártyaszám="1174865851185111" vevőID="1">
        <kártyán_szereplő_név>Nyíri Levente</kártyán_szereplő_név>
        <lejárat_i_dátum>12/23</lejárat_i_dátum>
        <biztonsági_kód>111</biztonsági_kód>
        <típus>Mastercard</típus>
    </Bankkártya>
    <Bankkártya kártyaszám="1176834769216856" vevőID="2">
        <kártyán_szereplő_név>Nagyné Kovács Rebeka</kártyán_szereplő_név>
        <lejárat_i_dátum>12/23</lejárat_i_dátum>
        <biztonsági_kód>111</biztonsági_kód>
        <típus>Visa</típus>
    </Bankkártya>
    <Bankkártya kártyaszám="1179863452169933" vevőID="2">
        <kártyán_szereplő_név>Nagy Balázs</kártyán_szereplő_név>
        <lejárat_i_dátum>02/27</lejárat_i_dátum>
        <biztonsági_kód>986</biztonsági_kód>
        <típus>Visa</típus>
    </Bankkártya>
    <Bankkártya kártyaszám="1175453336278779" vevőID="3">
        <kártyán_szereplő_név>Kis István</kártyán_szereplő_név>
        <lejárat_i_dátum>01/29</lejárat_i_dátum>
        <biztonsági_kód>494</biztonsági_kód>
        <típus>Mastercard</típus>
    </Bankkártya>
    <Bankkártya kártyaszám="1171154896532478" vevőID="4">
        <kártyán_szereplő_név>Varga Iván</kártyán_szereplő_név>
        <lejárat_i_dátum>01/24</lejárat_i_dátum>
        <biztonsági_kód>555</biztonsági_kód>
        <típus>Visa</típus>
    </Bankkártya>
    <!--Vevők-->
    <Vevő rendelésID="1" vevőID="1">
        <név>Nyíri Levente</név>
        <telefonszám>0620 589 9938</telefonszám>
        <cím>
            <irányítószám>3528</irányítószám>
            <közterület_neve>Csokonai Vitéz Mihály</közterület_neve>
            <közterület_megnevezése>utca</közterület_megnevezése>
            <házszám>11</házszám>
        </cím>
    </Vevő>
    <Vevő rendelésID="2" vevőID="2">
        <név>Nagy Balázs</név>

```

```

        <telefonszám>0630 597 6413</telefonszám>
        <cím>
            <irányítószám>3535</irányítószám>
            <közterület_neve>Kuruc</közterület_neve>
            <közterület_megnevezése>utca</közterület_megnevezése>
            <házzsám>23</házzsám>
            <emelet_ajtó>5/1</emelet_ajtó>
        </cím>
    </Vevő>
    <Vevő rendelésID="3" vevőID="3">
        <név>Kis István</név>
        <telefonszám>0620 365 5876</telefonszám>
        <cím>
            <irányítószám>3534</irányítószám>
            <közterület_neve>Nagy Lajos király</közterület_neve>
            <közterület_megnevezése>út</közterület_megnevezése>
            <házzsám>30</házzsám>
        </cím>
    </Vevő>
    <Vevő rendelésID="4" vevőID="4">
        <név>Varga Iván</név>
        <telefonszám>0630 879 1298</telefonszám>
        <cím>
            <irányítószám>3531</irányítószám>
            <közterület_neve>Aba</közterület_neve>
            <közterület_megnevezése>utca</közterület_megnevezése>
            <házzsám>2</házzsám>
            <emelet_ajtó>3/4</emelet_ajtó>
        </cím>
    </Vevő>
</F023QC_rendeletek>
Sikeres kiírás fájlba!

```

2.b) adatmódosítás

Ebben a feladatban 5 darab módosítást végeztem el az XML dokumentumon. Mindegyik módosításhoz külön függvényt hoztam létre.

- Étterem címének a módosítása
 - A 2-es étteremID-vel rendelkező étterem címét a következőre módosítottam:
3525 Miskolc Szemere Bertalan Utca 12.
 - A módosító függvénynek meg kell adni az étterem azonosítóját és azt, hogy mire szeretnék módosítani a címet

- Futár elérhetővé/nem elérhetővé tétele
 - Az egyes id-vel rendelkező futárt tettem elérhetővé
 - A függvénynek csak a futár azonosítóját kell megadni, mivel automatikusan állítja/változtatja a jelenlegi érték alapján azt, hogy a futár elérhető-e vagy sem
 - *elérhető* értéke:
 - 0, ha a futár elérhető
 - 1, ha a futár nem elérhető

- Rendeléshez tartozó termék mennyiségének a módosítása
 - Az 1-es rendelésben lévő 5-ös id-vel rendelkező termék mennyiségét módosítottam 2-re
 - A függvénynek meg kell adni a rendelésID-t és a termékID-t, valamint azt, hogy mennyire akarjuk módosítani a mennyiséget. A kettő id-re azért van szükség, hogy be tudjuk azonosítani az RT elemet, amelyben a mennyiséget szeretnénk módosítani
- Vevő házszaának módosítása
 - Az 1-es id-vel rendelkező vevő házszaát módosítottam 22-re
 - A függvénynek meg kell adni az azonosítót, valamint az új házszaot
- Vevő bankkártyáinak a típusainak a módosítása
 - Az 1-es id-vel rendelkező vevő bankkártyáinak a típusát állítottam be *MasterCard*-ra
 - A függvénynek csak a vevő azonosítóját kell megadni, hiszem az összes hozzá tartozó kártyának a típusát módosítja

Forráskód:

```
package hu.dompars.F023QC;
```

```
import java.io.File;
```

```
import java.io.FileWriter;
```

```
import java.io.IOException;
```

```
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.*;
```

```
import org.xml.sax.SAXException;
```

```
public class DOMModifyF023QC {

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

        File xmlFile = new File("XMLF023QC.xml");

        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();

        DocumentBuilder dBuilder = factory.newDocumentBuilder();

        Document document = dBuilder.parse(xmlFile);

        document.getDocumentElement().normalize();

        //Étterem címének a módosítása

        System.out.println("Második Étterem címének a módosítása");

        setEtterem(document, "2", "3525 Miskolc Szemere Bertalan Utca 12.");

        //Futár elérhetőségének a módosítása

        System.out.println("Első Futár elérhetőségének a módosítása");

        changeElerheto(document, "1");

        //Mennyiség módosítása

        System.out.println("Az első rendelésbe tartozó 5-ös id-vel rendelkező termék mennyiségének a módosítása");

        setMennyiseg(document, "1", "5", "2");

        //Házzám módosítása

        System.out.println("Az első vevő címében lévő házzám módosítása");

        setHazzsam(document, "1", "22");

        //A vevő összes kártyatípusának a módosítása

        System.out.println("Az első vevő bankkártyáinak beállítása MasterCard típusúra");

        setMasterCardforVevo(document, "1");

        //Dokumentum mentése és kiírása
```

```
writeXML(document);  
}
```

```
//Étterem címének a módosítása
```

```
private static void setEtterem(Document doc,String id,String cim) {
```

```
//Kiszedem a dokumentumból az éttermeket
```

```
NodeList ettermek = doc.getElementsByTagName("Étterem");
```

```
//Végigmegyek az éttermeken és ha egyezik a rendelésid a megadott id-vel, akkor az étterem címét átírom
```

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

```
if(ettermek.item(i).getAttributes().getNamedItem("étteremID").getTextContent().equalsIgnoreCase(id)) {
```

```
//Aktuális étterem
```

```
Element etterem = (Element) ettermek.item(i);
```

```
//Aktuális étterem előtte
```

```
System.out.println("Előtte:\n");
```

```
printNode(etterem);
```

```
//Cím kiválasztása és módosítása
```

```
Node cimNode = etterem.getElementsByTagName("cím").item(0);
```

```
cimNode.setTextContent(cim);
```

```
//Aktuális étterem utána
```

```
System.out.println("Utána:\n");
```

```
printNode(etterem);
```

```
}}}
```

```
//Futár elérhetőségének a módosítása
```

```
private static void changeElerheto(Document doc,String futarID) {
```

```
//A dokumentumból kiszedem az összes futárt, majd végigmegyek rajtuk egy for ciklussal
```

```
NodeList futarok = doc.getElementsByTagName("Futár");
```

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

```
//Aktuális futár
```

```

Element futar = (Element) futarok.item(i);

//Ha az aktuális futár id-je megegyezik a megadott id-vel
if(futar.getAttribute("futárID").equalsIgnoreCase(futarID)) {

    System.out.println("Előtte:\n");

    printNode(futar);

    //Elérhető nevezésű elem kiválasztása az aktuális futárból
    Element elerheto = (Element) futar.getElementsByTagName("elérhető").item(0);

    //Ha elérhető, akkor legyen nem elérhető
    if(elerheto.getTextContent().equalsIgnoreCase("1")) {

        elerheto.setTextContent("0");

        //Ha nem elérhető, akkor legyen elérhető
    }else {

        elerheto.setTextContent("1");

    }

    System.out.println("Utána:\n");

    printNode(futar);

}

}

//Mennyiség beállítása
private static void setMennyiség(Document doc,String rendelesID,String termékID,String mennyiség) {

    //Kiszedem a dokumentumból az összes RT-t
    NodeList RT = doc.getElementsByTagName("RT");

    //For ciklussal végigmegyem az összes RT-n
    for(int i =0;i<RT.getLength();i++) {

        //Az aktuális RT kapcsolat elem
        Element rt = (Element) RT.item(i);

        //RendelésId és termék id-k ellenőrzése
        if(rt.getAttribute("rendelésID").equalsIgnoreCase(rendelesID) &&
        rt.getAttribute("termékID").equalsIgnoreCase(termékID)) {

            System.out.println("Előtte:\n");

```

```
printNode(rt);

//Mennyiség elem kiválasztása és módosítása

Element menny = (Element) rt.getElementsByTagName("mennyiség").item(0);

menny.setTextContent(mennyiseg);

System.out.println("Utána:\n");

printNode(rt);

}}}
```

```
//Cím beállítása

private static void setHazszam(Document doc,String vevoID,String hazszam) {

//Vevők kigyűjtése

NodeList Vevok = doc.getElementsByTagName("Vevő");

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

//Aktuális vevő

Element vevo = (Element) Vevok.item(i);

//Vevőid ellenőrzése

if(vevo.getAttribute("vevőID").equalsIgnoreCase(vevoID)) {

System.out.println("Előtte:\n");

printNode(vevo);

//Cím elem kiválasztása

Element cim = (Element) vevo.getElementsByTagName("cím").item(0);

//Házzsám kiválasztása és beállítása

Element hazsz = (Element) cim.getElementsByTagName("házzsám").item(0);

hazsz.setTextContent(hazszam);

System.out.println("Utána:\n");

printNode(vevo);

}}}
```

```
//Bankkártyák beállítása

private static void setMasterCardforVevo(Document doc, String vevoID) {
```



```
//Bankkártyák kigyűjtése
```

```
NodeList kartyak = doc.getElementsByTagName("Bankkártya");
```

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

```
    //Aktuális bankkártya
```

```
    Element kartya = (Element) kartyak.item(i);
```

```
    //VevőID ellenőrzése
```

```
    if(kartya.getAttribute("vevőID").equalsIgnoreCase(vevoID)) {
```

```
        System.out.println("Előtte:\n");
```

```
        printNode(kartya);
```

```
        //típus kiválasztása és módosítása
```

```
        Element tipus = (Element) kartya.getElementsByTagName("típus").item(0);
```

```
        tipus.setTextContent("MasterCard");
```

```
        System.out.println("Utána:\n");
```

```
        printNode(kartya);
```

```
    } }
```

```
private static void removeText(Node root) {
```

```
    //Kigyűjtöm a routnak a gyerekeit, majd végigmegyek rajta
```

```
    NodeList nodeList = root.getChildNodes();
```

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

```
        //Ha talállok felesleget, akkor kitörölöm azt
```

```
        if (nodeList.item(i).getNodeType() == Node.TEXT_NODE &&  
            nodeList.item(i).getTextContent().strip().isEmpty()) {
```

```
            nodeList.item(i).getParentNode().removeChild(nodeList.item(i));
```

```
        //A törlés miatt 1-el előrébb járunk, mint ahol kéne, ezért az i-t csökkentem egyel
```

```
        i--;
```

```
        //Ha nem felesleges node, akkor ezt a node-ot is leellenőrzöm és kitörölöm belőle a felesleget
```

```
    } else {
```

```
        removeText(nodeList.item(i));
```

```
}}}
```

```
private static void writeXML(Document doc) {  
  
    try {  
  
        //</text> node-októl megtisztítom a dokumentumot, majd a transformer segítségével kiíratom egy fájlba  
        removeText(doc);  
  
        TransformerFactory transformerFactory = TransformerFactory.newInstance();  
        Transformer transformer = transformerFactory.newTransformer();  
        transformer.setOutputProperty(OutputKeys.ENCODING, "UTF-8");  
        transformer.setOutputProperty(OutputKeys.INDENT, "yes");  
        transformer.setOutputProperty("{https://xml.apache.org/xslt}indent-amount", "4");  
        FileWriter fileWriter = new FileWriter("XMLF023QC1.xml");  
        transformer.transform(new DOMSource(doc), new StreamResult(fileWriter));  
        transformer.transform(new DOMSource(doc), new StreamResult(System.out));  
        System.out.println("Sikeres kiírás fájlba!");  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

```
//Kiíratjuk a consolra a node értékeit
```

```
private static void printNode(Node currentElement) {  
  
    //Csak akkor fusson le, ha a típusa ELEMENT  
    if (currentElement.getNodeType() == Node.ELEMENT_NODE) {  
  
        //Első sor az xml szintaktikája + a node neve  
        String fRow = "<" + currentElement.getNodeName() + " ";  
  
        //Az első sorhoz hozzáadom az attribútumokat  
        for(int j = 0; j < currentElement.getAttributes().getLength(); j++) {  
  
            if(j == currentElement.getAttributes().getLength() - 1) {  
  
                fRow +=  
                    currentElement.getAttributes().item(j).getNodeName() + "=" + "\"" + currentElement.getAttributes().item(j).getTextContent() + "\">";  
            }  
        }  
    }  
}
```

```

}else {

fRow +=
currentElement.getAttributes().item(j).getNodeName()+"=\""+currentElement.getAttributes().item(j).getTextCo
ntent()+"\" ";

}}

//Kiíratom az első sort

System.out.println(getIndent(1)+fRow);

//Végigmegyek a Node gyerekelemein

NodeList childs = currentElement.getChildNodes();

for(int j=0;j<childs.getLength();j++) {

Node child = childs.item(j);

//Ha a gyerekelem típusa ELEMENT

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

//Ha csak sima gyerekelem, akkor kiíratom (Csak szöveg van benne)

if(child.getChildNodes().getLength()==1) {

System.out.println(getIndent(2)+"<"+child.getNodeName()+">"+child.getTextContent()+"</"+child.getNodeNa
me()+">");

//Ha nem csak szöveg van benne, akkor meg kell nézni a gyerekelem gyerekelemeit és azokat is kiíratni (cím
esetében)

}else {

//Kiírom a gyerekelemet

System.out.println(getIndent(2)+"<"+child.getNodeName()+">");

//Kiírom a gyerekelem gyerekelemeit

for(int k =0;k<child.getChildNodes().getLength();k++) {

if(child.getChildNodes().item(k).getNodeType()==Node.ELEMENT_NODE) {

System.out.println(getIndent(3)+"<"+child.getChildNodes().item(k).getNodeName()+">"+child.getChildNodes(
).item(k).getTextContent()+"</"+child.getChildNodes().item(k).getNodeName()+">");

}}

//Bezárom a gyerekelemet

System.out.println(getIndent(2)+"</"+child.getNodeName()+">");

```

```

}}}

//Bezárom az elemet

System.out.println(getIndent(1)+"</"+currentElement.getNodeName()+">\n");

}}

//Behúzások meghatározása

private static String getIndent(int indent) {

//A paraméternek megfelelő behúzást ad vissza, a behúzás 4 darab space

String ret="";

for(int i=0;i<indent;i++) {

ret+=" ";

}

return ret;

}}

```

Output:

Második Étterem címének a módosítása
Előtte:

```

<Étterem étteremID="2">
  <cím>3525 Miskolc Szemere Bertalan Utca 3.</cím>
  <nyitvatartás>Hétfő 7:00-18:00</nyitvatartás>
  <nyitvatartás>Kedd 7:00-18:00</nyitvatartás>
  <nyitvatartás>Szerda 7:00-18:00</nyitvatartás>
  <nyitvatartás>Csütörtök 7:00-18:00</nyitvatartás>
  <nyitvatartás>Péntek 7:00-18:00</nyitvatartás>
  <nyitvatartás>Szombat 8:00-18:00</nyitvatartás>
  <nyitvatartás>Vasárnap 10:00-15:00</nyitvatartás>
  <elérhetőség>https://falankfanny.hu/</elérhetőség>
</Étterem>

```

Utána:

```

<Étterem étteremID="2">
  <cím>3525 Miskolc Szemere Bertalan Utca 12.</cím>
  <nyitvatartás>Hétfő 7:00-18:00</nyitvatartás>
  <nyitvatartás>Kedd 7:00-18:00</nyitvatartás>
  <nyitvatartás>Szerda 7:00-18:00</nyitvatartás>
  <nyitvatartás>Csütörtök 7:00-18:00</nyitvatartás>
  <nyitvatartás>Péntek 7:00-18:00</nyitvatartás>
  <nyitvatartás>Szombat 8:00-18:00</nyitvatartás>
  <nyitvatartás>Vasárnap 10:00-15:00</nyitvatartás>
  <elérhetőség>https://falankfanny.hu/</elérhetőség>
</Étterem>

```

Első Futár elérhetőségének a módosítása

Előtte:

```
<Futár futárID="1" étteremID="1">
  <név>Nagy Milán</név>
  <telefonszám>0630 320 4870</telefonszám>
  <elérhető>0</elérhető>
  <típus>Biciklis</típus>
</Futár>
```

Utána:

```
<Futár futárID="1" étteremID="1">
  <név>Nagy Milán</név>
  <telefonszám>0630 320 4870</telefonszám>
  <elérhető>1</elérhető>
  <típus>Biciklis</típus>
</Futár>
```

Az első rendelésbe tartozó 5-ös id-vel rendelkező termék mennyiségének a módosítása

Előtte:

```
<RT rendelésID="1" termékID="5">
  <mennyiség>1</mennyiség>
  <nk_összetevő>Porcukor</nk_összetevő>
</RT>
```

Utána:

```
<RT rendelésID="1" termékID="5">
  <mennyiség>2</mennyiség>
  <nk_összetevő>Porcukor</nk_összetevő>
</RT>
```

Az első vevő címében lévő házszám módosítása

Előtte:

```
<Vevő rendelésID="1" vevőID="1">
  <név>Nyíri Levente</név>
  <telefonszám>0620 589 9938</telefonszám>
  <cím>
    <irányítószám>3528</irányítószám>
    <közterület_neve>Csokonai Vitéz Mihály</közterület_neve>
    <közterület_megnevezése>utca</közterület_megnevezése>
    <házszám>11</házszám>
  </cím>
</Vevő>
```

Utána:

```
<Vevő rendelésID="1" vevőID="1">
  <név>Nyíri Levente</név>
  <telefonszám>0620 589 9938</telefonszám>
  <cím>
    <irányítószám>3528</irányítószám>
    <közterület_neve>Csokonai Vitéz Mihály</közterület_neve>
    <közterület_megnevezése>utca</közterület_megnevezése>
    <házszám>22</házszám>
  </cím>
</Vevő>
```

Az első vevő bankkártyáinak beállítása MasterCard típusúra
Előtte:

```
<Bankkártya kártyaszám="1174859659985563" vevőID="1">
  <kártyán_szureplő_név>Nyíri Levente</kártyán_szureplő_név>
  <lejáratí_dátum>12/26</lejáratí_dátum>
  <biztonsági_kód>056</biztonsági_kód>
  <típus>Mastercard</típus>
</Bankkártya>
```

Utána:

```
<Bankkártya kártyaszám="1174859659985563" vevőID="1">
  <kártyán_szureplő_név>Nyíri Levente</kártyán_szureplő_név>
  <lejáratí_dátum>12/26</lejáratí_dátum>
  <biztonsági_kód>056</biztonsági_kód>
  <típus>MasterCard</típus>
</Bankkártya>
```

Előtte:

```
<Bankkártya kártyaszám="1174859655585888" vevőID="1">
  <kártyán_szureplő_név>Nyíri Levente</kártyán_szureplő_név>
  <lejáratí_dátum>05/28</lejáratí_dátum>
  <biztonsági_kód>356</biztonsági_kód>
  <típus>Visa</típus>
</Bankkártya>
```

Utána:

```
<Bankkártya kártyaszám="1174859655585888" vevőID="1">
  <kártyán_szureplő_név>Nyíri Levente</kártyán_szureplő_név>
  <lejáratí_dátum>05/28</lejáratí_dátum>
  <biztonsági_kód>356</biztonsági_kód>
  <típus>MasterCard</típus>
</Bankkártya>
```

Előtte:

```
<Bankkártya kártyaszám="1174865851185111" vevőID="1">
  <kártyán_szureplő_név>Nyíri Levente</kártyán_szureplő_név>
  <lejáratí_dátum>12/23</lejáratí_dátum>
  <biztonsági_kód>111</biztonsági_kód>
  <típus>Mastercard</típus>
</Bankkártya>
```

Utána:

```
<Bankkártya kártyaszám="1174865851185111" vevőID="1">
  <kártyán_szureplő_név>Nyíri Levente</kártyán_szureplő_név>
  <lejáratí_dátum>12/23</lejáratí_dátum>
  <biztonsági_kód>111</biztonsági_kód>
  <típus>MasterCard</típus>
</Bankkártya>
```

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<F023QC_rendelesek xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="XMLSchemaF023QC.xsd">
  <!--Rendelések-->
```

```
<Rendelés rendelésID="1" étteremID="1">
  <megjegyzés>A kutya harap!</megjegyzés>
  <rendelés_ideje>12:00</rendelés_ideje>
  <teljes_ár>15000</teljes_ár>
</Rendelés>
<Rendelés rendelésID="2" étteremID="1">
  <rendelés_ideje>14:32</rendelés_ideje>
  <teljes_ár>3000</teljes_ár>
</Rendelés>
<Rendelés rendelésID="3" étteremID="2">
  <megjegyzés>Ételalergia</megjegyzés>
  <rendelés_ideje>14:32</rendelés_ideje>
  <teljes_ár>3700</teljes_ár>
</Rendelés>
<Rendelés rendelésID="4" étteremID="3">
  <megjegyzés>Ételalergia</megjegyzés>
  <rendelés_ideje>14:32</rendelés_ideje>
  <teljes_ár>10000</teljes_ár>
</Rendelés>
<!--Éttermek-->
<Étterem étteremID="1">
  <cím>3528 Miskolc Csokonai Vitéz Mihály utca 40.</cím>
  <nyitvatartás>Hétfő 11:00-19:00</nyitvatartás>
  <nyitvatartás>Kedd 11:00-19:00</nyitvatartás>
  <nyitvatartás>Szerda 11:00-19:00</nyitvatartás>
  <nyitvatartás>Péntek 11:00-19:00</nyitvatartás>
  <elérhetőség>csarda@etterem.hu</elérhetőség>
</Étterem>
<Étterem étteremID="2">
  <cím>3525 Miskolc Szemere Bertalan Utca 12.</cím>
  <nyitvatartás>Hétfő 7:00-18:00</nyitvatartás>
  <nyitvatartás>Kedd 7:00-18:00</nyitvatartás>
  <nyitvatartás>Szerda 7:00-18:00</nyitvatartás>
  <nyitvatartás>Csütörtök 7:00-18:00</nyitvatartás>
  <nyitvatartás>Péntek 7:00-18:00</nyitvatartás>
  <nyitvatartás>Szombat 8:00-18:00</nyitvatartás>
  <nyitvatartás>Vasárnap 10:00-15:00</nyitvatartás>
  <elérhetőség>https://falankfanny.hu/</elérhetőség>
</Étterem>
<Étterem étteremID="3">
  <cím>3526 Miskolc, Petőfi Tér 1.</cím>
  <nyitvatartás>Hétfő 8:00-23:00</nyitvatartás>
  <nyitvatartás>Kedd 8:00-23:00</nyitvatartás>
  <nyitvatartás>Szerda 8:00-23:00</nyitvatartás>
  <nyitvatartás>Csütörtök 8:00-23:00</nyitvatartás>
  <nyitvatartás>Péntek 8:00-23:00</nyitvatartás>
  <elérhetőség>http://www.barkafood.hu/</elérhetőség>
</Étterem>
<!--Futárok-->
<Futár futárID="1" étteremID="1">
  <név>Nagy Milán</név>
  <telefonszám>0630 320 4870</telefonszám>
  <elérhető>1</elérhető>
  <típus>Biciklis</típus>
</Futár>
<Futár futárID="2" étteremID="1">
  <név>Kovács Aladár</név>
  <telefonszám>0630 560 8893</telefonszám>
  <elérhető>0</elérhető>
  <típus>Autós</típus>
```

```

</Futár>
<Futár futárID="3" étteremID="1">
  <név>Kiss Béla</név>
  <telefonszám>0630 632 4856</telefonszám>
  <elérhető>1</elérhető>
  <típus>Autós</típus>
</Futár>
<Futár futárID="4" étteremID="2">
  <név>Varga Roland</név>
  <telefonszám>0620 569 4236</telefonszám>
  <elérhető>1</elérhető>
  <típus>Rolleres</típus>
</Futár>
<Futár futárID="5" étteremID="2">
  <név>Kovács Péter</név>
  <telefonszám>0646 329 4568</telefonszám>
  <elérhető>0</elérhető>
  <típus>Biciklis</típus>
</Futár>
<Futár futárID="6" étteremID="3">
  <név>Szabó Éva</név>
  <telefonszám>0620 563 7821</telefonszám>
  <elérhető>1</elérhető>
  <típus>Autós</típus>
</Futár>
<!--Rendelés - Termék kapcsolat-->
<RT rendelésID="1" termékID="1">
  <menyiség>1</menyiség>
  <nk_összetevő>Répa</nk_összetevő>
</RT>
<RT rendelésID="1" termékID="3">
  <menyiség>4</menyiség>
</RT>
<RT rendelésID="1" termékID="5">
  <menyiség>2</menyiség>
  <nk_összetevő>Porcukor</nk_összetevő>
</RT>
<RT rendelésID="2" termékID="1">
  <menyiség>1</menyiség>
  <nk_összetevő>Zeller</nk_összetevő>
</RT>
<RT rendelésID="2" termékID="5">
  <menyiség>1</menyiség>
  <nk_összetevő>Porcukor</nk_összetevő>
</RT>
<RT rendelésID="3" termékID="2">
  <menyiség>5</menyiség>
  <nk_összetevő>Liszt</nk_összetevő>
  <nk_összetevő>Vanília Kivonat</nk_összetevő>
</RT>
<RT rendelésID="3" termékID="4">
  <menyiség>4</menyiség>
</RT>
<RT rendelésID="4" termékID="5">
  <menyiség>5</menyiség>
  <nk_összetevő>Só</nk_összetevő>
  <nk_összetevő>Porcukor</nk_összetevő>
  <nk_összetevő>Tejföl</nk_összetevő>
</RT>
<!--Termékek-->

```



```
<Termék termékID="1">
  <név>Húsleves</név>
  <ár>1000</ár>
  <elkészítési_idő>30 perc</elkészítési_idő>
  <összetevők>sárgarépa</összetevők>
  <összetevők>fehérrépa</összetevők>
  <összetevők>karalábé</összetevők>
  <összetevők>zeller</összetevők>
  <összetevők>vöröshagyma</összetevők>
  <összetevők>paprika</összetevők>
  <összetevők>petrezselyem</összetevők>
  <összetevők>só</összetevők>
  <összetevők>bors</összetevők>
  <összetevők>tészta</összetevők>
</Termék>
<Termék termékID="2">
  <név>Brownie</név>
  <ár>500</ár>
  <elkészítési_idő>15 perc</elkészítési_idő>
  <összetevők>liszt</összetevők>
  <összetevők>só</összetevők>
  <összetevők>kakaópor</összetevők>
  <összetevők>étcsokoládé</összetevők>
  <összetevők>vaj</összetevők>
  <összetevők>nutella</összetevők>
  <összetevők>tojás</összetevők>
  <összetevők>cukor</összetevők>
  <összetevők>vanília kivonat</összetevők>
</Termék>
<Termék termékID="3">
  <név>Omlós libacomb</név>
  <ár>4000</ár>
  <elkészítési_idő>130 perc</elkészítési_idő>
  <összetevők>libacomb</összetevők>
  <összetevők>tengeri só</összetevők>
</Termék>
<Termék termékID="4">
  <név>Muffin</név>
  <ár>300</ár>
  <elkészítési_idő>30 perc</elkészítési_idő>
  <összetevők>tojás</összetevők>
  <összetevők>cukor</összetevők>
  <összetevők>vaj</összetevők>
  <összetevők>tej</összetevők>
  <összetevők>liszt</összetevők>
  <összetevők>só</összetevők>
  <összetevők>sütőpor</összetevők>
  <összetevők>vaníliás cukor</összetevők>
</Termék>
<Termék termékID="5">
  <név>Túrógombóc</név>
  <ár>2000</ár>
  <elkészítési_idő>25 perc</elkészítési_idő>
  <összetevők>túró</összetevők>
  <összetevők>tojás</összetevők>
  <összetevők>búzadara</összetevők>
  <összetevők>só</összetevők>
  <összetevők>szódabikarbóna</összetevők>
  <összetevők>zsemlemorzsa</összetevők>
  <összetevők>olaj</összetevők>
```

```

        <összetevők>porcukor</összetevők>
        <összetevők>tejföl</összetevők>
</Termék>
<!--Bankkártyák-->
<Bankkártya kártyaszám="1174859659985563" vevőID="1">
    <kártyán_szereplő_név>Nyíri Levente</kártyán_szereplő_név>
    <lejárat_i_dátum>12/26</lejárat_i_dátum>
    <biztonsági_kód>056</biztonsági_kód>
    <típus>MasterCard</típus>
</Bankkártya>
<Bankkártya kártyaszám="1174859655585888" vevőID="1">
    <kártyán_szereplő_név>Nyíri Levente</kártyán_szereplő_név>
    <lejárat_i_dátum>05/28</lejárat_i_dátum>
    <biztonsági_kód>356</biztonsági_kód>
    <típus>MasterCard</típus>
</Bankkártya>
<Bankkártya kártyaszám="1174865851185111" vevőID="1">
    <kártyán_szereplő_név>Nyíri Levente</kártyán_szereplő_név>
    <lejárat_i_dátum>12/23</lejárat_i_dátum>
    <biztonsági_kód>111</biztonsági_kód>
    <típus>MasterCard</típus>
</Bankkártya>
<Bankkártya kártyaszám="1176834769216856" vevőID="2">
    <kártyán_szereplő_név>Nagyné Kovács Rebeka</kártyán_szereplő_név>
    <lejárat_i_dátum>12/23</lejárat_i_dátum>
    <biztonsági_kód>111</biztonsági_kód>
    <típus>Visa</típus>
</Bankkártya>
<Bankkártya kártyaszám="1179863452169933" vevőID="2">
    <kártyán_szereplő_név>Nagy Balázs</kártyán_szereplő_név>
    <lejárat_i_dátum>02/27</lejárat_i_dátum>
    <biztonsági_kód>986</biztonsági_kód>
    <típus>Visa</típus>
</Bankkártya>
<Bankkártya kártyaszám="1175453336278779" vevőID="3">
    <kártyán_szereplő_név>Kis István</kártyán_szereplő_név>
    <lejárat_i_dátum>01/29</lejárat_i_dátum>
    <biztonsági_kód>494</biztonsági_kód>
    <típus>Mastercard</típus>
</Bankkártya>
<Bankkártya kártyaszám="1171154896532478" vevőID="4">
    <kártyán_szereplő_név>Varga Iván</kártyán_szereplő_név>
    <lejárat_i_dátum>01/24</lejárat_i_dátum>
    <biztonsági_kód>555</biztonsági_kód>
    <típus>Visa</típus>
</Bankkártya>
<!--Vevők-->
<Vevő rendelésID="1" vevőID="1">
    <név>Nyíri Levente</név>
    <telefonszám>0620 589 9938</telefonszám>
    <cím>
        <irányítószám>3528</irányítószám>
        <közterület_neve>Csokonai Vitéz Mihály</közterület_neve>
        <közterület_megnevezése>utca</közterület_megnevezése>
        <házszám>22</házszám>
    </cím>
</Vevő>
<Vevő rendelésID="2" vevőID="2">
    <név>Nagy Balázs</név>
    <telefonszám>0630 597 6413</telefonszám>

```

```

        <cím>
            <irányítószám>3535</irányítószám>
            <közterület_neve>Kuruc</közterület_neve>
            <közterület_megnevezése>utca</közterület_megnevezése>
            <házzsám>23</házzsám>
            <emelet_ajtó>5/1</emelet_ajtó>
        </cím>
    </Vevő>
    <Vevő rendelésID="3" vevőID="3">
        <név>Kis István</név>
        <telefonszám>0620 365 5876</telefonszám>
        <cím>
            <irányítószám>3534</irányítószám>
            <közterület_neve>Nagy Lajos király</közterület_neve>
            <közterület_megnevezése>út</közterület_megnevezése>
            <házzsám>30</házzsám>
        </cím>
    </Vevő>
    <Vevő rendelésID="4" vevőID="4">
        <név>Varga Iván</név>
        <telefonszám>0630 879 1298</telefonszám>
        <cím>
            <irányítószám>3531</irányítószám>
            <közterület_neve>Aba</közterület_neve>
            <közterület_megnevezése>utca</közterület_megnevezése>
            <házzsám>2</házzsám>
            <emelet_ajtó>3/4</emelet_ajtó>
        </cím>
    </Vevő>
</F023QC_rendeletek>
Sikeres kiírás fájlba!

```

2.c) Adatlekérdezés

Ebben a feladatban is függvényekkel oldottam meg az egyes lekérdezéseket. Az alábbi lekérdezéseket hoztam létre:

- Az összes elérhető futár lekérdezése
- Az összes bankkártya lekérdezése, amelynek a típusa MasterCard
- Az összes rendelt termék lekérdezése
- Egy étteremhez tartozó rendelések lekérdezése
- Az egy irányítószámról származó rendelések lekérdezése

Forráskód:

```
package hu.dompars.F023QC;

import java.io.File;

import java.io.IOException;

import java.util.ArrayList;

import java.util.List;

import javax.xml.parsers.DocumentBuilder;

import javax.xml.parsers.DocumentBuilderFactory;

import javax.xml.parsers.ParserConfigurationException;


import org.w3c.dom.*;

import org.xml.sax.SAXException;


public class DOMQueryF023QC {

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

        File xmlFile = new File("XMLF023QC.xml");

        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();

        DocumentBuilder dBuilder = factory.newDocumentBuilder();

        Document document = dBuilder.parse(xmlFile);

        document.getDocumentElement().normalize();


        //Lekérdezések

        //Lekérdezem az összes elérhető futárt

        System.out.println("Az elérhető futárok:");

        getElerheto(document);


        //Lekérdezem azokat a bankkártyákat, amelyeknek a típusa mastercard

        System.out.println();

        System.out.println("MasterCard Bankkártyák");
```

```
getMaster(document);
```

```
//Lekérdezem a rendelésben lévő termékeket
```

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

```
System.out.println("Rendelésben lévő termékek:");
```

```
getTermekByRendeles(document, "1");
```

```
//Lekérdezem az étteremhez tartozó rendeléseket
```

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

```
System.out.println("Étteremhez tartozó rendelések:");
```

```
getRendelesByEtterem(document, "1");
```

```
//Lekérdezem azokat a rendeléseket, amelyek ugyanarról a környékről származnak (irányítószám alapján)
```

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

```
System.out.println("Rendelések egy környékről:");
```

```
getRendelesByIrszam(document, "3531");
```

```
}
```

```
private static void getRendelesByIrszam(Document doc, String irszam) {
```

```
//Kigyűjtjük az összes vevőt
```

```
NodeList vevok = doc.getElementsByTagName("Vevő");
```

```
List<String> rendelesIDk = new ArrayList<String>();
```

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

```
//Ez a jelenlegi vevő
```

```
Element vevo = (Element) vevok.item(i);
```

```
//A jelenlegi vevőnek a címe
```

```
Element cim = (Element) vevo.getElementsByTagName("cím").item(0);
```

```
//A jelenlegi irányítószám
```

```
Element iranyitoszam = (Element) cim.getElementsByTagName("irányítószám").item(0);
```

```
//Ha a jelenlegi irányítószám megegyezik a megadottal, akkor a vevőnek a rendelés id-jét hozzáadjuk a listához  
if(iranyitoszam.getTextContent().equalsIgnoreCase(irszam)) {  
    rendelesIDk.add(vevo.getAttribute("rendelésID"));  
}
```

```
//A dokumentumból kiszedem a rendeléseket  
NodeList rendelesek = doc.getElementsByTagName("Rendelés");
```

```
//végigmegyek az összes rendelésen  
for(int i =0;i<rendelesek.getLength();i++) {  
  
    //Ha a rendelés id-je benne van az irányítószámhoz tartozó rendelésid-k listájában, akkor kiíratom  
    if(rendelesIDk.contains(rendelesek.item(i).getAttributes().getNamedItem("rendelésID").getTextContent())) {  
        printNode(rendelesek.item(i));  
    }  
}
```

```
private static void getRendelesByEtterem(Document doc, String etteremID) {
```

```
    //A dokumentumból kiszedem az összes rendelést
```

```
    NodeList Rendelesek = doc.getElementsByTagName("Rendelés");
```

```
    //Végigmegyek a rendeléseken és ha egyezik a megadott étteremid-vel az attribute értéke, akkor kiíratom
```

```
    for(int i=0;i<Rendelesek.getLength();i++) {  
  
        if(Rendelesek.item(i).getAttributes().getNamedItem("étteremID").getTextContent().equalsIgnoreCase(etteremID)) {  
            printNode(Rendelesek.item(i));  
        }  
    }
```

```
private static void getTermekByRendeles(Document doc,String rendelesID) {
```

```
    //Lekérem az összes RT kapcsolatot
```

```
    NodeList RT = doc.getElementsByTagName("RT");
```

```
    //Ebben fpgom tárolni a termékek id-it
```

```
    List<String> termekIDk = new ArrayList<String>();
```

```
//végigmegyek a kapcsolat elemein és megnézem a rendelésID-ket, ha egyezik akkor a termék id-t hozzáadom a listához
```

```
for(int i=0;i < RT.getLength();i++) {  
    if(RT.item(i).getAttributes().getNamedItem("rendelésID").getTextContent().equalsIgnoreCase(rendelesID)) {  
        termékIDk.add(RT.item(i).getAttributes().getNamedItem("termékID").getTextContent());  
    }  
}
```

```
//Kigyűjtöm az összes terméket a dokumentumból
```

```
NodeList termekek = doc.getElementsByTagName("Termék");
```

```
//Végigmegyek a termékeken és ha egyezést találok, akkor kiíratom a terméket
```

```
for(int i =0;i<termekek.getLength();i++) {  
    //Ha benne a van a listában a termék id-je, akkor benne van a rendelésben  
    if(termékIDk.contains(termekek.item(i).getAttributes().getNamedItem("termékID").getTextContent())) {  
        printNode(termekek.item(i));  
    }  
}
```

```
private static void getElerheto(Document doc) {
```

```
//Kigyűjtöm a futárokat a dokumentumból
```

```
NodeList futarok = doc.getElementsByTagName("Futár");
```

```
//Végigmegyek a futárokon
```

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

```
//Kigyűjtöm az aktuális futár gyerekelemeit
```

```
NodeList childs = futarok.item(i).getChildNodes();
```

```
//Végigmegyek a gyerekelemeken és keresem az elérhető nevű gyerekelemet
```

```
for(int j=0;j<childs.getLength();j++) {
```

```
//Ha a futár elérhető, akkor kiíratom
```

```
if(childs.item(j).getNodeName().equalsIgnoreCase("elérhető") &&  
    childs.item(j).getTextContent().equalsIgnoreCase("1") ) {
```

```

printNode(futarok.item(i));

System.out.println();

}}}}

private static void getMaster(Document doc) {

//Kigyűjtöm a kártyákat a dokumentumból

NodeList kartyak = doc.getElementsByTagName("Bankkártya");


//Végigmegyek a kártyákon

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

//Az aktuális kártya gyerekelemei

NodeList childs = kartyak.item(i).getChildNodes();


//Megkeresem a típus nevű gyerekelemet

for(int j=0;j<childs.getLength();j++) {

//Ha az értéke Mastercard, akkor kiíratom a kártyát

if(childs.item(j).getNodeName().equalsIgnoreCase("típus") &&
childs.item(j).getTextContent().equalsIgnoreCase("Mastercard") ) {

printNode(kartyak.item(i));

System.out.println();

}}}}


//Kiíratjuk a consolra a node értékeit

private static void printNode(Node currentElement) {

//Csak akkor fusson le, ha a típusa ELEMENT

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

//Első sor az xml szintaktikája + a node neve

String fRow ="<" +currentElement.getNodeName()+" ";


//Az első sorhoz hozzáadom az attribútumokat

for(int j = 0; j< currentElement.getAttributes().getLength();j++) {

if(j==currentElement.getAttributes().getLength()-1) {

```



```

fRow +=
currentElement.getAttributes().item(j).getNodeName()+"=\""+currentElement.getAttributes().item(j).getTextCo
ntent()+"\">";

}else {

fRow +=
currentElement.getAttributes().item(j).getNodeName()+"=\""+currentElement.getAttributes().item(j).getTextCo
ntent()+"\" ";

}

}

//Kiíratom az első sort

System.out.println(getIndent(1)+fRow);


//Végigmegyek a Node gyerekelemein

NodeList childs = currentElement.getChildNodes();

for(int j=0;j<childs.getLength();j++) {

Node child = childs.item(j);

//Ha a gyerekelem típusa ELEMENT

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

//Ha csak sima gyerekelem, akkor kiíratom (Csak szöveg van benne)

if(child.getChildNodes().getLength()==1) {

System.out.println(getIndent(2)+"<"+child.getNodeName()+">"+child.getTextContent()+"</"+child.getNodeNa
me()+">");

//Ha nem csak szöveg van benne, akkor meg kell nézni a gyerekelem gyerekelemeit és azokat is kiíratni (cím
esetében)

}else {

//Kiírom a gyerekelemet

System.out.println(getIndent(2)+"<"+child.getNodeName()+">");

//Kiírom a gyerekelem gyerekelemeit

for(int k =0;k<child.getChildNodes().getLength();k++) {

if(child.getChildNodes().item(k).getNodeType()==Node.ELEMENT_NODE) {

System.out.println(getIndent(3)+"<"+child.getChildNodes().item(k).getNodeName()+">"+child.getChildNodes(
).item(k).getTextContent()+"</"+child.getChildNodes().item(k).getNodeName()+">");

}}

//Bezárom a gyerekelemet

```

```

System.out.println(getIndent(2)+"</"+child.getNodeName()+">");

}}}

//Bezárom az elemet

System.out.println(getIndent(1)+"</"+currentElement.getNodeName()+">\n");

}}

//Behúzások meghatározása

private static String getIndent(int indent) {

//A paraméternek megfelelő behúzást ad vissza, a behúzás 4 darab space

String ret="";

for(int i=0;i<indent;i++) {

ret+=" ";

}

return ret;

}}

```

Output:

Az elérhető futárok:

```

<Futár futárID="3" étteremID="1">
  <név>Kiss Béla</név>
  <telefonszám>0630 632 4856</telefonszám>
  <elérhető>1</elérhető>
  <típus>Autós</típus>
</Futár>

<Futár futárID="4" étteremID="2">
  <név>Varga Roland</név>
  <telefonszám>0620 569 4236</telefonszám>
  <elérhető>1</elérhető>
  <típus>Rolleres</típus>
</Futár>

<Futár futárID="6" étteremID="3">
  <név>Szabó Éva</név>
  <telefonszám>0620 563 7821</telefonszám>
  <elérhető>1</elérhető>
  <típus>Autós</típus>
</Futár>

```

MasterCard Bankkártyák

```

<Bankkártya kártyaszám="1174859659985563" vevőID="1">

```

```
<kártyán_szureplő_név>Nyíri Levente</kártyán_szureplő_név>
<lejárat_i_dátum>12/26</lejárat_i_dátum>
<biztonsági_kód>056</biztonsági_kód>
<típus>Mastercard</típus>
</Bankkártya>
```

```
<Bankkártya kártyaszám="1174865851185111" vevőID="1">
  <kártyán_szureplő_név>Nyíri Levente</kártyán_szureplő_név>
  <lejárat_i_dátum>12/23</lejárat_i_dátum>
  <biztonsági_kód>111</biztonsági_kód>
  <típus>Mastercard</típus>
</Bankkártya>
```

```
<Bankkártya kártyaszám="1175453336278779" vevőID="3">
  <kártyán_szureplő_név>Kis István</kártyán_szureplő_név>
  <lejárat_i_dátum>01/29</lejárat_i_dátum>
  <biztonsági_kód>494</biztonsági_kód>
  <típus>Mastercard</típus>
</Bankkártya>
```

Rendelésben lévő termékek:

```
<Termék termékID="1">
  <név>Húsleves</név>
  <ár>1000</ár>
  <elkészítési_idő>30 perc</elkészítési_idő>
  <összetevők>sárgarépa</összetevők>
  <összetevők>fehérrépa</összetevők>
  <összetevők>karalábé</összetevők>
  <összetevők>zeller</összetevők>
  <összetevők>vöröshagyma</összetevők>
  <összetevők>paprika</összetevők>
  <összetevők>petrezselyem</összetevők>
  <összetevők>só</összetevők>
  <összetevők>bors</összetevők>
  <összetevők>tészta</összetevők>
</Termék>
```

```
<Termék termékID="3">
  <név>Omlós libacomb</név>
  <ár>4000</ár>
  <elkészítési_idő>130 perc</elkészítési_idő>
  <összetevők>libacomb</összetevők>
  <összetevők>tengeri só</összetevők>
</Termék>
```

```
<Termék termékID="5">
  <név>Túrógombóc</név>
  <ár>2000</ár>
  <elkészítési_idő>25 perc</elkészítési_idő>
  <összetevők>túró</összetevők>
  <összetevők>tojás</összetevők>
  <összetevők>búzadara</összetevők>
  <összetevők>só</összetevők>
  <összetevők>szódabikarbóna</összetevők>
  <összetevők>zsemlemorzsa</összetevők>
  <összetevők>olaj</összetevők>
```

```
        <összetevők>porcukor</összetevők>
        <összetevők>tejföl</összetevők>
    </Termék>
```

Étteremhez tartozó rendelések:

```
    <Rendelés rendelésID="1" étteremID="1">
        <megjegyzés>A kutya harap!</megjegyzés>
        <rendelés_ideje>12:00</rendelés_ideje>
        <teljes_ár>15000</teljes_ár>
    </Rendelés>
```

```
    <Rendelés rendelésID="2" étteremID="1">
        <rendelés_ideje>14:32</rendelés_ideje>
        <teljes_ár>3000</teljes_ár>
    </Rendelés>
```

Rendelések egy környékről:

```
    <Rendelés rendelésID="4" étteremID="3">
        <megjegyzés>Ételalergia</megjegyzés>
        <rendelés_ideje>14:32</rendelés_ideje>
        <teljes_ár>10000</teljes_ár>
    </Rendelés>
```

2.d) Adatírás

Ebben a feladatban egy új XML dokumentumot hoztam létre, melynek megadtam gyökérelemét és a gyökérelemben található elemeket. Mindegyik elemnek van egy saját függvénye, amellyel hozzá lehet adni a gyökérelemhez, ezekkel építettem fel a fa struktúrát. A vevőben található cím gyerekelemhez külön metódust hoztam létre, amely Node típusú tér vissza, ezt a függvényt a vevő létrehozásakor használom. Az új dokumentumhoz hozzáadtam még kommenteket.

Forráskód:

```
package hu.domparse.F023QC;
```

```
import javax.xml.parsers.DocumentBuilder;
```

```
import javax.xml.parsers.DocumentBuilderFactory;
```

```
import javax.xml.transform.OutputKeys;
```

```
import javax.xml.transform.Transformer;
```

```
import javax.xml.transform.TransformerFactory;
```

```
import javax.xml.transform.dom.DOMSource;
```

```
import javax.xml.transform.stream.StreamResult;
```

```
import org.w3c.dom.Document;
```

```
import org.w3c.dom.Element;
```

```
import org.w3c.dom.Node;
```

```
import java.io.FileWriter;
```

```
import java.util.ArrayList;
```

```
import java.util.Arrays;
```

```
import java.util.List;
```

```
public class DOMWriteF023QC {
```

```
    public static void main(String[] args) {
```

```
        try {
```

```
            //Új XML fájl létrehozása
```

```
            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
```

```
            DocumentBuilder dBuilder = factory.newDocumentBuilder();
```

```
            Document newDocument = dBuilder.newDocument();
```

```
                //A gyökérelem létrehozása és hozzáadása a dokumentumhoz
```

```
            Element rootElement = newDocument.createElement("F023QC_rendelesek");
```

```
            rootElement.setAttribute("xmlns:xsi", "http://www.w3.org/2001/XMLSchema-instance");
```

```
            rootElement.setAttribute("xsi:noNamespaceSchemaLocation", "XMLSchemaF023QC.xsd");
```

```
            newDocument.appendChild(rootElement);
```

```
            //Rendelések hozzáadása a kommenttel
```

```
            rootElement.appendChild(newDocument.createComment("Rendelések"));
```

```
            addRendeles(newDocument, rootElement, "1","1","A kutya harap!","12:00","15000");
```

```
addRendeles(newDocument, rootElement, "2","1","", "14:32", "3000");  
addRendeles(newDocument, rootElement, "3","2", "Ételalergia", "14:32", "3700");  
addRendeles(newDocument, rootElement, "4","3", "Ételalergia", "14:32", "10000");
```

//Éttermek hozzáadása a kommenttel

```
rootElement.appendChild(newDocument.createComment("Éttermek"));  
  
addEtterem(newDocument, rootElement, "1", "3528 Miskolc Csokonai Vitéz Mihály utca  
40.", "Hétfő 11:00-19:00,Kedd 11:00-19:00,Szerda 11:00-19:00,Péntek 11:00-  
19:00", "csarda@etterem.hu");  
  
addEtterem(newDocument, rootElement, "2", "3525 Miskolc Szemere Bertalan Utca  
3.", "Hétfő 7:00-18:00,Kedd 7:00-18:00,Szerda 7:00-18:00,Csütörtök 7:00-18:00,Péntek 7:00-  
18:00,Szombat 8:00-18:00,Vasárnap 10:00-15:00", "https://falankfanny.hu/");  
  
addEtterem(newDocument, rootElement, "3", "3526 Miskolc, Petőfi Tér 1.", "Hétfő 8:00-  
23:00,Kedd 8:00-23:00,Szerda 8:00-23:00,Csütörtök 8:00-23:00,Péntek 8:00-  
23:00", "http://www.barkafood.hu/");
```

//Futárok hozzáadása a kommenttel

```
rootElement.appendChild(newDocument.createComment("Futárok"));  
  
addFutar(newDocument, rootElement, "1","1", "Nagy Milán", "0630 320  
4870", "0", "Biciklis");  
  
addFutar(newDocument, rootElement, "2","1", "Kovács Aladár", "0630 560  
8893", "0", "Autós");  
  
addFutar(newDocument, rootElement, "3","1", "Kiss Béla", "0630 632 4856", "1", "Autós");  
  
addFutar(newDocument, rootElement, "4","2", "Varga Roland", "0620 569  
4236", "1", "Rolleres");  
  
addFutar(newDocument, rootElement, "5","2", "Kovács Péter", "0646 329  
4568", "0", "Biciklis");  
  
addFutar(newDocument, rootElement, "6","3", "Szabó Éva", "0620 563 7821", "1", "Autós");
```

//Több-több kapcsolat hozzáadása a kommenttel

```
rootElement.appendChild(newDocument.createComment("Rendelés - Termék kapcsolat"));  
addRT(newDocument, rootElement, "1","1","1", "Répa");
```

```

addRT(newDocument, rootElement, "1","3","4","");
addRT(newDocument, rootElement, "1","5","1","Porcukor");
addRT(newDocument, rootElement, "2","1","1","Zeller");
addRT(newDocument, rootElement, "2","5","1","Porcukor");
addRT(newDocument, rootElement, "3","2","5","Liszt,Vanília Kivonat");
addRT(newDocument, rootElement, "3","4","4","");
addRT(newDocument, rootElement, "4","5","5","Só,Porcukor,Tejföl");

//Termékek hozzáadása a kommenttel

rootElement.appendChild(newDocument.createComment("Termékek"));

addTermek(newDocument, rootElement, "1","Húsleves","1000","30
perc","sárgarépa,fehérrépa,karalábé,zeller,vöröshagyma,paprika,petrezselyem,só,bors,tészta");

addTermek(newDocument, rootElement, "2","Brownie","500","15
perc","liszt,só,kakaópor,écsokoládé,vaj,nutella,tojás,cukor,vanília kivonat");

addTermek(newDocument, rootElement, "3","Omlós libacomb","4000","130
perc","libacomb,tengeri só");

addTermek(newDocument, rootElement, "4","Muffin","300","30
perc","tojás,cukor,vaj,tej,liszt,só,sütőpor,vaníliás cukor");

addTermek(newDocument, rootElement, "5","Túrógombóc","2000","25
perc","túró,tojás,búzadara,só,szódabikarbóna,zsemlemorzsa,olaj,porcukor,tejföl");

//Bankkártyák hozzáadása a kommenttel

rootElement.appendChild(newDocument.createComment("Bankkártyák"));

addKartya(newDocument, rootElement, "1174859659985563","1","Nyíri
Levente","12/26","056","Mastercard");

addKartya(newDocument, rootElement, "1174859655585888","1","Nyíri
Levente","05/28","356","Visa");

addKartya(newDocument, rootElement, "1174865851185111","1","Nyíri
Levente","12/23","111","Mastercard");

addKartya(newDocument, rootElement, "1176834769216856","2","Nagyné Kovács
Rebeka","12/23","111","Visa");

```

```
addKartya(newDocument, rootElement, "1179863452169933", "2", "Nagy  
Balázs", "02/27", "986", "Visa");
```

```
addKartya(newDocument, rootElement, "1175453336278779", "3", "Kis  
István", "01/29", "494", "Mastercard");
```

```
addKartya(newDocument, rootElement, "1171154896532478", "4", "Varga  
Iván", "01/24", "555", "Visa");
```

```
//Vevők hozzáadása a kommenttel, valamint a cím létrehozása és hozzáadása a vevőhöz
```

```
rootElement.appendChild(newDocument.createComment("Vevők"));
```

```
addVevo(newDocument, rootElement, "1", "1", "Nyíri Levente", "0620 589  
9938", addCim(newDocument, "3528", "Csokonai Vitéz Mihály", "utca", "11", ""));
```

```
addVevo(newDocument, rootElement, "2", "2", "Nagy Balázs", "0630 597  
6413", addCim(newDocument, "3535", "Kuruc", "utca", "23", "5/1"));
```

```
addVevo(newDocument, rootElement, "3", "3", "Kis István", "0620 365  
5876", addCim(newDocument, "3534", "Nagy Lajos király", "út", "30", ""));
```

```
addVevo(newDocument, rootElement, "4", "4", "Varga Iván", "0630 879  
1298", addCim(newDocument, "3531", "Aba", "utca", "2", "3/4"));
```

```
//Kiírás fájlba és a konzolra
```

```
TransformerFactory transformerFactory = TransformerFactory.newInstance();
```

```
Transformer transformer = transformerFactory.newTransformer();
```

```
transformer.setOutputProperty(OutputKeys.ENCODING, "UTF-8");
```

```
transformer.setOutputProperty(OutputKeys.INDENT, "yes");
```

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

```
FileWriter fileWriter = new FileWriter("XMLF023QC1.xml");
```

```
transformer.transform(new DOMSource(newDocument), new StreamResult(fileWriter));
```

```
transformer.transform(new DOMSource(newDocument), new StreamResult(System.out
```

```
 } catch (Exception e) {
```

```
System.out.println("Hiba történt: " + e.getMessage());
```



```
}}
```

```
//Függvények, amelyekkel új elemeket hozhatunk létre
```

```
//Rendelés létrehozása
```

```
private static void addRendeles(Document document, Element rootElement,String  
rendelesID,String etteremID,String megjegyz,String ido,String ar) {
```

```
//Létrehozom a rendelést és beállítom az attribútumait
```

```
Element rendeles = document.createElement("Rendelés");
```

```
rendeles.setAttribute("rendelésID", rendelesID );
```

```
rendeles.setAttribute("étteremID", etteremID);
```

```
//Létrhozom a gyerekelemeket, majd beállítom a nevüket, ezután beállítom az értéküket,  
majd hozzáadom az újonnan létrehozott rendeléshez
```

```
//A rendelésnek nem feltétlenül kell tartalmazi megjegyzést, ezért ha azt nem adunk meg,  
akkor nem hozunk létre gyerekelemet
```

```
if(!megjegyz.equalsIgnoreCase("")) {
```

```
Element megjegyzes = document.createElement("megjegyzés");
```

```
megjegyzes.setTextContent(megjegyz);
```

```
rendeles.appendChild(megjegyzes);
```

```
}
```

```
Element ideje = document.createElement("rendelés_ideje");
```

```
ideje.setTextContent(ido);
```

```
rendeles.appendChild(ideje);
```

```
Element teljesar = document.createElement("teljes_ár");
```

```
teljesar.setTextContent(ar);
```

```
rendeles.appendChild(teljesar);
```

```
//Hozzáadjuk a rendelést a gyökérelemhez
```

```
rootElement.appendChild(rendeles);
```

```
}
```

```
//Étterem létrehozása
```

```
private static void addEtterem(Document document, Element rootElement,String  
etteremID,String cimin,String nyitva,String elerhet) {
```

```
//Létrehozom az éttermet és beállítom az attribútumait
```

```
Element etterem = document.createElement("Étterem");
```

```
etterem.setAttribute("etteremID", etteremID);
```

```
//Létrhozom a gyerekelemeket, majd beállítom a nevüket, ezután beállítom az értéküket,  
majd hozzáadom az újonnan létrehozot étteremhez
```

```
Element cim = document.createElement("cím");
```

```
cim.setTextContent(cimin);
```

```
etterem.appendChild(cim);
```

```
//A nyitvatartás többször is előfordulhat az étterem elemben Stringként adom meg ,-vel  
elválasztva, ezért először listává alakítom
```

```
List<String> nyitvatartasok = new ArrayList<String>(Arrays.asList(nyitva.split(",")));
```

```
//Végigmegyek a listán, és az értékeknek megfelelően létrehozom a nyitvatartást
```

```
for (String string : nyitvatartasok) {
```

```
Element nyitvat = document.createElement("nyitvatartás");
```

```
nyitvat.setTextContent(string);
```

```
etterem.appendChild(nyitvat);
```

```
}
```

```
Element elerhetoseg = document.createElement("elérhetőség");
elerhetoseg.setTextContent(elerhet);
etterem.appendChild(elerhetoseg);
```

```
//Hozzáadom az éttermet a gyökérelemhez
```

```
rootElement.appendChild(etterem);
}
```

```
//Futár létrehozása
```

```
private static void addFutar(Document document, Element rootElement,String futarID,String
etteremID,String nevin,String telefon,String elerhet,String tip) {
```

```
//Létrehozom a futárt és beállítom az attribútumait
```

```
Element futar = document.createElement("Futár");
futar.setAttribute("futarID", futarID);
futar.setAttribute("étteremID", etteremID);
```

```
//Létrhozom a gyerekelemeket, majd beállítom a nevüket, ezután beállítom az értéküket,
majd hozzáadom az újonnan létrehozott futárhoz
```

```
Element nev = document.createElement("név");
nev.setTextContent(nevin);
futar.appendChild(nev);
```

```
Element tel = document.createElement("telefonszám");
tel.setTextContent(telefon);
futar.appendChild(tel);
```

```
Element elerheto = document.createElement("elérhető");
```

```
elerheto.setTextContent(elerhet);
```

```
futar.appendChild(elerheto);
```

```
Element tipus = document.createElement("típus");
```

```
tipus.setTextContent(tip);
```

```
futar.appendChild(tipus);
```

```
//hozzáadom a futárt a gyökérelemhez
```

```
rootElement.appendChild(futar);
```

```
}
```

```
//N:M kapcsolat létrehozása
```

```
private static void addRT(Document document, Element rootElement,String  
rendelesID,String termékID,String mennyiseg,String NK) {
```

```
//Létrehozom a kapcsolat elemét és beállítom az attribútumait ezek alapján lehet  
beazonosítani, hogy melyik rendelésben milyen termék szerepel
```

```
Element rt = document.createElement("RT");
```

```
rt.setAttribute("rendelésID", rendelesID);
```

```
rt.setAttribute("termékID", termékID);
```

```
//Létrehozom a gyerekelemeket, majd beállítom a nevüket, ezután beállítom az értéküket,  
majd hozzáadom az újonnan létrehozott RT kapcsolathoz
```

```
Element menny = document.createElement("mennyiség");
```

```
menny.setTextContent(mennyiseg);
```

```
rt.appendChild(menny);
```

```
//A nem kívánt összetevő olyan gyerekelem, amelynek nincs megszabva, hogy hányszor  
fordulhat elő,(0,1,2,3...) ezért először ellenőrzöm, hogy akarunk e hozzáadni ilyen  
gyerekelemet az RT-hez
```

```
if(!NK.equalsIgnoreCase("")) {
```

//Ha akarunk, akkor a ,-k mentén listává alakítom, majd végigmegyek az értékeken és létrehozom a gyerekelemeket

```
List<String> nkk = new ArrayList<String>(Arrays.asList(NK.split(", ")));
```

```
for (String string : nkk) {
```

```
    Element nk = document.createElement("nk_összetevő");
```

```
    nk.setTextContent(string);
```

```
    rt.appendChild(nk);
```

```
}
```

//hozzáadom a gyökérelemhez

```
rootElement.appendChild(rt);
```

```
}
```

//Termék létrehozása

```
private static void addTermek(Document document, Element rootElement, String  
termekID, String neve, String ara, String elkido, String ossze) {
```

//Létrehozom a terméket és beállítom az attribútumait

```
Element termék = document.createElement("Termék");
```

```
termék.setAttribute("termékID", termékID);
```

//Létrehozom a gyerekelemeket, majd beállítom a nevüket, ezután beállítom az értéküket, majd hozzáadom az újonnan létrehozott termékhez

```
Element nev = document.createElement("név");
```

```
nev.setTextContent(neve);
```

```
termék.appendChild(nev);
```

```
Element ar = document.createElement("ár");
```

```
ar.setTextContent(ara);
```

```
termék.appendChild(ar);
```

```
Element ido = document.createElement("elkészítési_idő");
```

```
ido.setTextContent(elkido);
```

```
termek.appendChild(ido);
```

//Az összetevő többször is előfordulhat a termék elemben Stringként adom meg ,-vel elválasztva, ezért először listává alakítom

```
List<String> osszetevok = new ArrayList<String>(Arrays.asList(ossze.split(",")));
```

//Ezután az értékeknek megfelelően létrehozom az elemeket

```
for (String string : osszetevok) {
```

```
Element osszetevo = document.createElement("összetevők");
```

```
osszetevo.setTextContent(string);
```

```
termek.appendChild(osszetevo);
```

```
}
```

//Hozzáadom a gyökérelemhez

```
rootElement.appendChild(termek);
```

```
}
```

//Kártya létrehozása

```
private static void addKartya(Document document, Element rootElement,String szam,String  
vevoID,String nevin,String lejaratidatum,String biztkod,String tip) {
```

//Létrehozom a kártyát és beállítom az attribútumait

```
Element kartya = document.createElement("Bankkártya");
```

```
kartya.setAttribute("kártyaszám", szam);
```

```
kartya.setAttribute("vevőID", vevoID);
```

//Létrehozom a gyerekelemeket, majd beállítom a nevüket, ezután beállítom az értéküket, majd hozzáadom az újonnan létrehozott kártyához

```
Element nev = document.createElement("kártyán_szerelő_név");
```

```
nev.textContent(nevin);
```

```
kartya.appendChild(nev);
```

```
Element datum = document.createElement("lejárati_dátum");
```

```
datum.textContent(lejaratidatum);
```

```
kartya.appendChild(datum);
```

```
Element kod = document.createElement("biztonsági_kód");
```

```
kod.textContent(biztkod);
```

```
kartya.appendChild(kod);
```

```
Element tipus = document.createElement("típus");
```

```
tipus.textContent(tip);
```

```
kartya.appendChild(tipus);
```

//Hozzáadom a kártyát a gyökérelemhez

```
rootElement.appendChild(kartya);
```

```
}
```

//Cím létrehozása

//Nem void a visszatérési érték, hanem Node, mivel ezt majd hozzá fogjuk adni a vevőhöz

```
private static Node addCim(Document document,String iranyzam,String kneve,String  
kneve,String hazsz,String emelet) {
```

//Létrehozom a címet, ezzel fogok visszatérni

```
Element cim = document.createElement("cím");
```

```
//Létrhozom a gyerekelemeket, majd beállítom a nevüket, ezután beállítom az értéküket
```

```
Element irszam = document.createElement("írányítószám");
```

```
irszam.setTextContent(iranyszam);
```

```
cim.appendChild(irszam);
```

```
Element neve = document.createElement("közterület_neve");
```

```
neve.setTextContent(kneve);
```

```
cim.appendChild(neve);
```

```
Element megnevezes = document.createElement("közterület_megnevezése");
```

```
megnevezes.setTextContent(kmegnev);
```

```
cim.appendChild(megnevezes);
```

```
Element hazszam = document.createElement("házszám");
```

```
hazszam.setTextContent(hasz);
```

```
cim.appendChild(hazszam);
```

```
//Nem biztos,hogy a címben szerepel, hogy emelet/ajtó, ezért leellenőrzöm
```

```
if(!emelet.equalsIgnoreCase("")) {
```

```
Element ajto = document.createElement("emelet_ajtó");
```

```
ajto.setTextContent(emelet);
```

```
cim.appendChild(ajto);
```

```
}
```

```
//Visszatérek a Node-al
```

```
return cim;
```

```
}
```



```

//Vevő létrehozása

//A cím itt már Node-ként jön át, így azt csak hozzá kell adnunk a vevőhöz

private static void addVevo(Document document, Element rootElement,String vevoID,String
rendelesID,String neve,String telefon,Node cimin) {

//Létrehozom a vevőt és beállítom az attribútumait

Element vevo = document.createElement("Vevő");

vevo.setAttribute("vevőID", vevoID);

vevo.setAttribute("rendelésID", rendelesID);


//Létrehozom a gyerekelemeket, majd beállítom a nevüket, ezután beállítom az értéküket,
majd hozzáadom az újonnan létrehozott vevőhöz

Element nev = document.createElement("név");

nev.setTextContent(neve);

vevo.appendChild(nev);


Element tel = document.createElement("telefonszám");

tel.setTextContent(telefon);

vevo.appendChild(tel);


//Hozzáadom a címet a vevőhöz

vevo.appendChild(cimin);

//Hozzáadom a vevőt a gyökérelemhez

rootElement.appendChild(vevo);

}

}

```

Output:

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<F023QC_rendelesek xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="XMLSchemaF023QC.xsd">

```

```
<!--Rendelések-->
<Rendelés rendelésID="1" étteremID="1">
  <megjegyzés>A kutya harap!</megjegyzés>
  <rendelés_ideje>12:00</rendelés_ideje>
  <teljes_ár>15000</teljes_ár>
</Rendelés>
<Rendelés rendelésID="2" étteremID="1">
  <rendelés_ideje>14:32</rendelés_ideje>
  <teljes_ár>3000</teljes_ár>
</Rendelés>
<Rendelés rendelésID="3" étteremID="2">
  <megjegyzés>Ételalergia</megjegyzés>
  <rendelés_ideje>14:32</rendelés_ideje>
  <teljes_ár>3700</teljes_ár>
</Rendelés>
<Rendelés rendelésID="4" étteremID="3">
  <megjegyzés>Ételalergia</megjegyzés>
  <rendelés_ideje>14:32</rendelés_ideje>
  <teljes_ár>10000</teljes_ár>
</Rendelés>
<!--Éttermek-->
<Étterem étteremID="1">
  <cím>3528 Miskolc Csokonai Vitéz Mihály utca 40.</cím>
  <nyitvatartás>Hétfő 11:00-19:00</nyitvatartás>
  <nyitvatartás>Kedd 11:00-19:00</nyitvatartás>
  <nyitvatartás>Szerda 11:00-19:00</nyitvatartás>
  <nyitvatartás>Péntek 11:00-19:00</nyitvatartás>
  <elérhetőség>csarda@etterem.hu</elérhetőség>
</Étterem>
<Étterem étteremID="2">
  <cím>3525 Miskolc Szemere Bertalan Utca 3.</cím>
  <nyitvatartás>Hétfő 7:00-18:00</nyitvatartás>
  <nyitvatartás>Kedd 7:00-18:00</nyitvatartás>
  <nyitvatartás>Szerda 7:00-18:00</nyitvatartás>
  <nyitvatartás>Csütörtök 7:00-18:00</nyitvatartás>
  <nyitvatartás>Péntek 7:00-18:00</nyitvatartás>
  <nyitvatartás>Szombat 8:00-18:00</nyitvatartás>
  <nyitvatartás>Vasárnap 10:00-15:00</nyitvatartás>
  <elérhetőség>https://falankfanny.hu/</elérhetőség>
</Étterem>
<Étterem étteremID="3">
  <cím>3526 Miskolc, Petőfi Tér 1.</cím>
  <nyitvatartás>Hétfő 8:00-23:00</nyitvatartás>
  <nyitvatartás>Kedd 8:00-23:00</nyitvatartás>
  <nyitvatartás>Szerda 8:00-23:00</nyitvatartás>
  <nyitvatartás>Csütörtök 8:00-23:00</nyitvatartás>
  <nyitvatartás>Péntek 8:00-23:00</nyitvatartás>
  <elérhetőség>http://www.barkafood.hu/</elérhetőség>
</Étterem>
<!--Futárok-->
<Futár futárID="1" étteremID="1">
  <név>Nagy Milán</név>
  <telefonszám>0630 320 4870</telefonszám>
  <elérhető>0</elérhető>
  <típus>Biciklis</típus>
</Futár>
<Futár futárID="2" étteremID="1">
  <név>Kovács Aladár</név>
  <telefonszám>0630 560 8893</telefonszám>
  <elérhető>0</elérhető>
```

```

        <típus>Autós</típus>
</Futár>
<Futár futárID="3" étteremID="1">
    <név>Kiss Béla</név>
    <telefonszám>0630 632 4856</telefonszám>
    <elérhető>1</elérhető>
    <típus>Autós</típus>
</Futár>
<Futár futárID="4" étteremID="2">
    <név>Varga Roland</név>
    <telefonszám>0620 569 4236</telefonszám>
    <elérhető>1</elérhető>
    <típus>Rolleres</típus>
</Futár>
<Futár futárID="5" étteremID="2">
    <név>Kovács Péter</név>
    <telefonszám>0646 329 4568</telefonszám>
    <elérhető>0</elérhető>
    <típus>Biciklis</típus>
</Futár>
<Futár futárID="6" étteremID="3">
    <név>Szabó Éva</név>
    <telefonszám>0620 563 7821</telefonszám>
    <elérhető>1</elérhető>
    <típus>Autós</típus>
</Futár>
<!--Rendelés - Termék kapcsolat-->
<RT rendelésID="1" termékID="1">
    <menyiség>1</menyiség>
    <nk_összetevő>Répa</nk_összetevő>
</RT>
<RT rendelésID="1" termékID="3">
    <menyiség>4</menyiség>
</RT>
<RT rendelésID="1" termékID="5">
    <menyiség>1</menyiség>
    <nk_összetevő>Porcukor</nk_összetevő>
</RT>
<RT rendelésID="2" termékID="1">
    <menyiség>1</menyiség>
    <nk_összetevő>Zeller</nk_összetevő>
</RT>
<RT rendelésID="2" termékID="5">
    <menyiség>1</menyiség>
    <nk_összetevő>Porcukor</nk_összetevő>
</RT>
<RT rendelésID="3" termékID="2">
    <menyiség>5</menyiség>
    <nk_összetevő>Liszt</nk_összetevő>
    <nk_összetevő>Vanília Kivonat</nk_összetevő>
</RT>
<RT rendelésID="3" termékID="4">
    <menyiség>4</menyiség>
</RT>
<RT rendelésID="4" termékID="5">
    <menyiség>5</menyiség>
    <nk_összetevő>Só</nk_összetevő>
    <nk_összetevő>Porcukor</nk_összetevő>
    <nk_összetevő>Tejföl</nk_összetevő>
</RT>

```

```
<!--Termékek-->
<Termék termékID="1">
  <név>Húsleves</név>
  <ár>1000</ár>
  <elkészítési_idő>30 perc</elkészítési_idő>
  <összetevők>sárgarépa</összetevők>
  <összetevők>fehérrépa</összetevők>
  <összetevők>karalábé</összetevők>
  <összetevők>zeller</összetevők>
  <összetevők>vöröshagyma</összetevők>
  <összetevők>paprika</összetevők>
  <összetevők>petrezselyem</összetevők>
  <összetevők>só</összetevők>
  <összetevők>bors</összetevők>
  <összetevők>tészta</összetevők>
</Termék>
<Termék termékID="2">
  <név>Brownie</név>
  <ár>500</ár>
  <elkészítési_idő>15 perc</elkészítési_idő>
  <összetevők>liszt</összetevők>
  <összetevők>só</összetevők>
  <összetevők>kakaópor</összetevők>
  <összetevők>étcsokoládé</összetevők>
  <összetevők>vaj</összetevők>
  <összetevők>nutella</összetevők>
  <összetevők>tojás</összetevők>
  <összetevők>cukor</összetevők>
  <összetevők>vanília kivonat</összetevők>
</Termék>
<Termék termékID="3">
  <név>Omlós libacomb</név>
  <ár>4000</ár>
  <elkészítési_idő>130 perc</elkészítési_idő>
  <összetevők>libacomb</összetevők>
  <összetevők>tengeri só</összetevők>
</Termék>
<Termék termékID="4">
  <név>Muffin</név>
  <ár>300</ár>
  <elkészítési_idő>30 perc</elkészítési_idő>
  <összetevők>tojás</összetevők>
  <összetevők>cukor</összetevők>
  <összetevők>vaj</összetevők>
  <összetevők>tej</összetevők>
  <összetevők>liszt</összetevők>
  <összetevők>só</összetevők>
  <összetevők>sütőpor</összetevők>
  <összetevők>vaníliás cukor</összetevők>
</Termék>
<Termék termékID="5">
  <név>Túrógombóc</név>
  <ár>2000</ár>
  <elkészítési_idő>25 perc</elkészítési_idő>
  <összetevők>túró</összetevők>
  <összetevők>tojás</összetevők>
  <összetevők>búzadara</összetevők>
  <összetevők>só</összetevők>
  <összetevők>szódabikarbóna</összetevők>
  <összetevők>zsemlemorzsa</összetevők>
```

```
        <összetevők>olaj</összetevők>
        <összetevők>porcukor</összetevők>
        <összetevők>tejföl</összetevők>
</Termék>
<!--Bankkártyák-->
<Bankkártya kártyaszám="1174859659985563" vevőID="1">
    <kártyán_szereplő_név>Nyíri Levente</kártyán_szereplő_név>
    <lejárat_i_dátum>12/26</lejárat_i_dátum>
    <biztonsági_kód>056</biztonsági_kód>
    <típus>Mastercard</típus>
</Bankkártya>
<Bankkártya kártyaszám="1174859655585888" vevőID="1">
    <kártyán_szereplő_név>Nyíri Levente</kártyán_szereplő_név>
    <lejárat_i_dátum>05/28</lejárat_i_dátum>
    <biztonsági_kód>356</biztonsági_kód>
    <típus>Visa</típus>
</Bankkártya>
<Bankkártya kártyaszám="1174865851185111" vevőID="1">
    <kártyán_szereplő_név>Nyíri Levente</kártyán_szereplő_név>
    <lejárat_i_dátum>12/23</lejárat_i_dátum>
    <biztonsági_kód>111</biztonsági_kód>
    <típus>Mastercard</típus>
</Bankkártya>
<Bankkártya kártyaszám="1176834769216856" vevőID="2">
    <kártyán_szereplő_név>Nagyné Kovács Rebeka</kártyán_szereplő_név>
    <lejárat_i_dátum>12/23</lejárat_i_dátum>
    <biztonsági_kód>111</biztonsági_kód>
    <típus>Visa</típus>
</Bankkártya>
<Bankkártya kártyaszám="1179863452169933" vevőID="2">
    <kártyán_szereplő_név>Nagy Balázs</kártyán_szereplő_név>
    <lejárat_i_dátum>02/27</lejárat_i_dátum>
    <biztonsági_kód>986</biztonsági_kód>
    <típus>Visa</típus>
</Bankkártya>
<Bankkártya kártyaszám="1175453336278779" vevőID="3">
    <kártyán_szereplő_név>Kis István</kártyán_szereplő_név>
    <lejárat_i_dátum>01/29</lejárat_i_dátum>
    <biztonsági_kód>494</biztonsági_kód>
    <típus>Mastercard</típus>
</Bankkártya>
<Bankkártya kártyaszám="1171154896532478" vevőID="4">
    <kártyán_szereplő_név>Varga Iván</kártyán_szereplő_név>
    <lejárat_i_dátum>01/24</lejárat_i_dátum>
    <biztonsági_kód>555</biztonsági_kód>
    <típus>Visa</típus>
</Bankkártya>
<!--Vevők-->
<Vevő rendelésID="1" vevőID="1">
    <név>Nyíri Levente</név>
    <telefonszám>0620 589 9938</telefonszám>
    <cím>
        <irányítószám>3528</irányítószám>
        <közterület_neve>Csokonai Vitéz Mihály</közterület_neve>
        <közterület_megnevezése>utca</közterület_megnevezése>
        <házszám>11</házszám>
    </cím>
</Vevő>
<Vevő rendelésID="2" vevőID="2">
    <név>Nagy Balázs</név>
```

```
<telefonszám>0630 597 6413</telefonszám>
<cím>
  <irányítószám>3535</irányítószám>
  <közterület_neve>Kuruc</közterület_neve>
  <közterület_megnevezése>utca</közterület_megnevezése>
  <házzsám>23</házzsám>
  <emelet_ajtó>5/1</emelet_ajtó>
</cím>
</Vevő>
<Vevő rendelésID="3" vevőID="3">
  <név>Kis István</név>
  <telefonszám>0620 365 5876</telefonszám>
  <cím>
    <irányítószám>3534</irányítószám>
    <közterület_neve>Nagy Lajos király</közterület_neve>
    <közterület_megnevezése>út</közterület_megnevezése>
    <házzsám>30</házzsám>
  </cím>
</Vevő>
<Vevő rendelésID="4" vevőID="4">
  <név>Varga Iván</név>
  <telefonszám>0630 879 1298</telefonszám>
  <cím>
    <irányítószám>3531</irányítószám>
    <közterület_neve>Aba</közterület_neve>
    <közterület_megnevezése>utca</közterület_megnevezése>
    <házzsám>2</házzsám>
    <emelet_ajtó>3/4</emelet_ajtó>
  </cím>
</Vevő>
</F023QC_rendeletek>
```