

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Éttermek kiszállításának adatbázisa

Készítette: **Palkó Patrik Dávid**

Neptunkód: **ZW7DOR**

A feladat leírása:

A beadandó témája egy olyan adatbázis, amely egyszerre több kiszállító éttermet kezel. Rákereshetünk benne az éttermekben dolgozó futárookra, a beszállítókra, Étteek mennyiségére, árára, vagy a vevő adataira

Az Adatbázisrendszerek I. című tárgy beadandó feladatomat helyeztem át XML-be

Az egyedek közötti kapcsolat

- **Étterem és Futár:**

Az Étterem és a Futár egyedek között egy a többhöz kapcsolat van, mivel egy étterem

alkalmazhat több futárt, de egy futár csak egy étteremnél dolgozik.

- **Étterem és Beszállító:**

Az Étterem és a Beszállító egyedek között több a többhöz kapcsolat van, mivel egy

étterem rendelhet több beszállítótól, valamint egy beszállító beszállíthat több

étteremnek is. A kapcsolat paraméterei: a Hozzávalók, amely a beszállító által

beszállított hozzávalókat jelenti, valamint a Dátum, azaz a beszállítás dátuma.

- **Étterem és Étel:**

Az Étterem és az Étel egyedek között egy a többhöz kapcsolat van, mivel egy

étteremnek lehet több étele is, de egy étel csak egy étteremhez tartozhat.

- **Étel és Vevő:**

Az Étel és a Vevő egyedek között több a többhöz kapcsolat van, mivel egy vevő

rendelhet többfajta ételt, és egy ételből rendelhet több különböző vevő is.

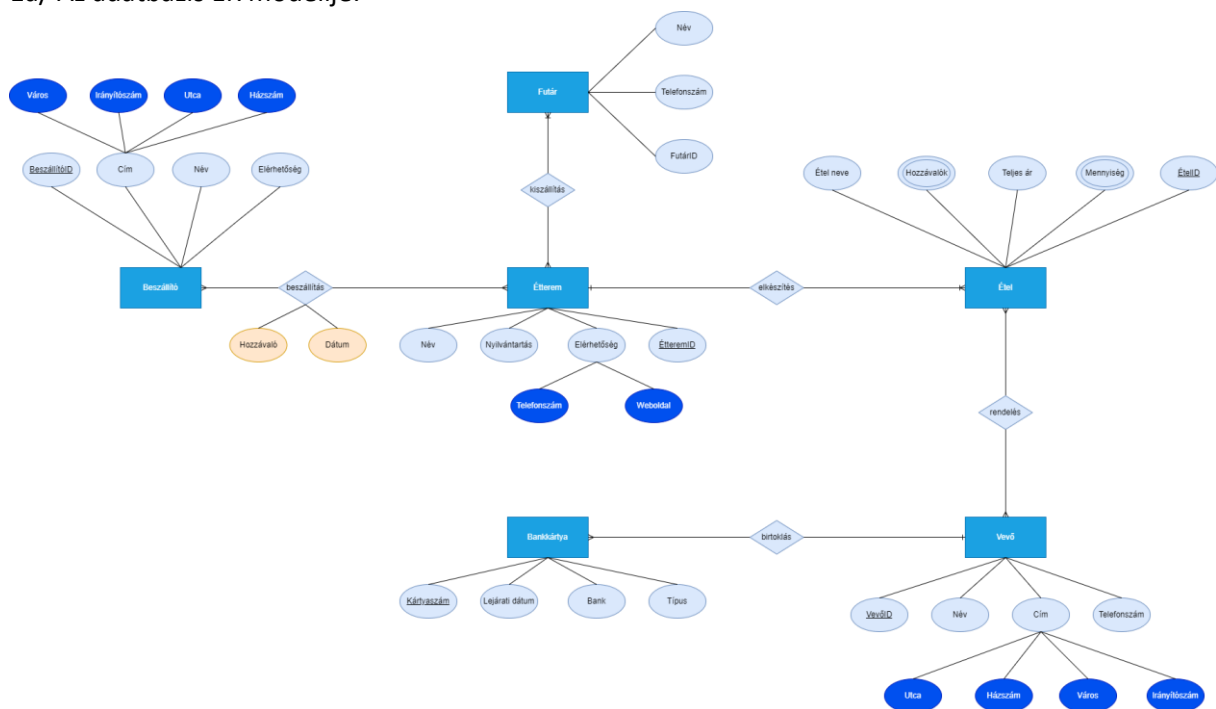
- **Vevő és Bankkártya:**

A Vevő és a Bankkártya egyedek között egy a többhöz kapcsolat van, mivel egy

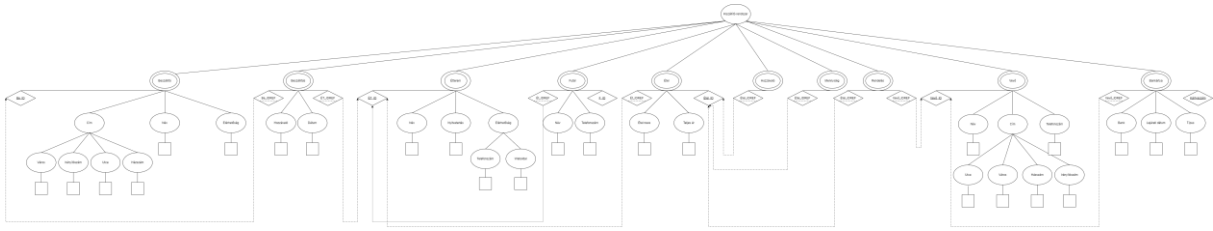
vevőnek lehet több bankkártyája is, de egy bankkártyának nem lehet több tulajdonosa

1.Feladat

1a/ Az adatbázis ER modellje:



1b/ Adatbázis konvertálása XDM modellre:



1c/ XDM alapján egy XML dokumentum készítése:

```
<?xml version="1.0" encoding="UTF-8"?>
<kiszallito_rendszer xmlns:xs="http://www.w3.org/2001/XMLSchema-instance" xs:noNamespaceSchemaLocation="XMLSchemaZW7DOR.xsd">

  <!--Etterem-->

  <etterem ET_ID="1">
    <Nyitvatartas>11től 22:30ig</Nyitvatartas>
    <Nev>Kalipszo</Nev>
    <Elerhetoseg>
      <Telefonszam>209876543</Telefonszam>
      <Weboldal>kalipszo.hu</Weboldal>
    </Elerhetoseg>
  </etterem>

  <etterem ET_ID="2">
    <Nyitvatartas>9től 21ig</Nyitvatartas>
    <Nev>Vendelin</Nev>
    <Elerhetoseg>
      <Telefonszam>208765432</Telefonszam>
      <Weboldal>vendelin.hu</Weboldal>
    </Elerhetoseg>
  </etterem>

  <etterem ET_ID="3">
    <Nyitvatartas>8től 22ig</Nyitvatartas>
    <Nev>Molnár Csárda</Nev>
    <Elerhetoseg>
      <Telefonszam>306876546</Telefonszam>
      <Weboldal>molnarcarda.hu</Weboldal>
    </Elerhetoseg>
  </etterem>


```

```
<!--Futar-->

<futar F_ID="1" ET_IDREF="1">
  <Nev>Nagy Tíbor</Nev>
  <Telefonszam>202354342</Telefonszam>
</futar>

<futar F_ID="2" ET_IDREF="2">
  <Nev>Horváth Lajos</Nev>
  <Telefonszam>203521958</Telefonszam>
</futar>

<futar F_ID="3" ET_IDREF="3">
  <Nev>Palkó Patrik</Nev>
  <Telefonszam>202572553</Telefonszam>
</futar>


```

```
<!--Beszallito-->

<beszallito Be_ID="1">
  <Nev>Kis Piac</Nev>
  <Elerhetoseg>kis.piac@gmail.com</Elerhetoseg>
  <Cim>
    <Varos>Kazincbarcika</Varos>
    <Iranyitoszam>3700</Iranyitoszam>
    <Utca>Fő utca</Utca>
    <Hatszam>35</Hatszam>
  </Cim>
</beszallito>

<beszallito Be_ID="2">
  <Nev>Nagy piac</Nev>
  <Elerhetoseg>nagy.piac@gmail.com</Elerhetoseg>
  <Cim>
    <Varos>Encs</Varos>
    <Iranyitoszam>3896</Iranyitoszam>
    <Utca>Fő utca</Utca>
    <Hatszam>25</Hatszam>
  </Cim>
</beszallito>

<beszallito Be_ID="3">
  <Nev>Közepes piac</Nev>
  <Elerhetoseg>közepes.piac@gmail.com</Elerhetoseg>
  <Cim>
    <Varos>Miskolc</Varos>
    <Iranyitoszam>3885</Iranyitoszam>
    <Utca>Fő utca</Utca>
    <Hatszam>15</Hatszam>
  </Cim>
</beszallito>
```

```
<!--Beszallitas-->

<beszallitas Be_IDREF="1" ET_IDREF="1">
  <Hozzavalo>banan</Hozzavalo>
  <Datum>2022.08.06</Datum>
</beszallitas>

<beszallitas Be_IDREF="2" ET_IDREF="2">
  <Hozzavalo>csoki</Hozzavalo>
  <Datum>2022.03.01</Datum>
</beszallitas>

<beszallitas Be_IDREF="3" ET_IDREF="3">
  <Hozzavalo>alma</Hozzavalo>
  <Datum>2022.10.21</Datum>
</beszallitas>

<!--Etel-->

<etel Etel_ID="1" ET_IDREF="1">
  <etel_neve>Banános palacsinta</etel_neve>
  <teljes_ar>2500</teljes_ar>
</etel>

<etel Etel_ID="2" ET_IDREF="2">
  <etel_neve>Csokis zserbó</etel_neve>
  <teljes_ar>3400</teljes_ar>
</etel>

<etel Etel_ID="3" ET_IDREF="3">
  <etel_neve>Disznó almával a szájában</etel_neve>
  <teljes_ar>25000</teljes_ar>
</etel>
```

```
<!--Vevő-->

<vevo vevo_ID="1">
  <Nev>Palkó Zsolt</Nev>
  <Telefonszam>202566311</Telefonszam>
  <Cim>
    <Utca>Károlyi Gáspár</Utca>
    <Varos>Szerencs</Varos>
    <Hazzsam>5</Hazzsam>
    <Iranyitoszam>3880</Iranyitoszam>
  </Cim>
</vevo>

<vevo vevo_ID="2">
  <Nev>Lakatos Valentin</Nev>
  <Telefonszam>202599210</Telefonszam>
  <Cim>
    <Utca>Kálmán</Utca>
    <Varos>Budapest</Varos>
    <Hazzsam>31</Hazzsam>
    <Iranyitoszam>3550</Iranyitoszam>
  </Cim>
</vevo>

<vevo vevo_ID="3">
  <Nev>Raul Patricio</Nev>
  <Telefonszam>202599901</Telefonszam>
  <Cim>
    <Utca>Nagy Lajos</Utca>
    <Varos>Pálháza</Varos>
    <Hazzsam>2</Hazzsam>
    <Iranyitoszam>3770</Iranyitoszam>
  </Cim>
</vevo>
```

```
<!--Bankártya-->

<bankartya kartyaszam="7629746286234247" vevo_IDREF="1">
  <Bank>OTP</Bank>
  <Lejarati_datum>2025.10.28</Lejarati_datum>
  <Tipus>betéti kártya</Tipus>
</bankartya>

<bankartya kartyaszam="(0543345349664954" vevo_IDREF="1">
  <Bank>MKB</Bank>
  <Lejarati_datum>2025.10.29</Lejarati_datum>
  <Tipus>hitelkártya</Tipus>
</bankartya>

<bankartya kartyaszam="5432467233423432" vevo_IDREF="1">
  <Bank>CIB Bank</Bank>
  <Lejarati_datum>2025.10.30</Lejarati_datum>
  <Tipus>többfunkciós kártya</Tipus>
</bankartya>

/kiszallito_rendszer>
```

1d/ XML alapján XMLSchema készítése:


```

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

  <xs:element name="kiszallito_rendszer">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="etterem" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Nyitvatartas" type="xs:string"/>
              <xs:element name="Nev" type="xs:string"/>
              <xs:element name="Elerhetoseg">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Telefonszam" type="xs:int"/>
                    <xs:element name="Weboldal" type="xs:string"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
            <xs:attribute name="ET_ID" type="xs:int"/>
          </xs:complexType>
        </xs:element>
        <xs:element name="futar" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Nev" type="xs:string"/>
              <xs:element name="Telefonszam" type="xs:int"/>
            </xs:sequence>
            <xs:attribute name="F_ID" type="xs:int"/>
            <xs:attribute name="ET_IDREF" type="xs:int"/>
          </xs:complexType>
        </xs:element>
        <xs:element name="beszallito" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Nev" type="xs:string"/>
              <xs:element name="Elerhetoseg" type="xs:string"/>
              <xs:element name="Cim">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Varos" type="xs:string"/>
                    <xs:element name="Iranditoszam" type="xs:int"/>
                    <xs:element name="Utca" type="xs:string"/>
                    <xs:element name="Hatszszam" type="xs:int"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```

```

        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="Be_ID" type="xs:int"/></xs:attribute>
</xs:complexType>
</xs:element>
<xs:element name="beszallitas" maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Hozzavalo" type="xs:string"/></xs:element>
      <xs:element name="Datum" type="xs:date"/></xs:element>
    </xs:sequence>
    <xs:attribute name="Be_IDREF" type="xs:int"/></xs:attribute>
    <xs:attribute name="ET_IDREF" type="xs:int"/></xs:attribute>
  </xs:complexType>
</xs:element>
<xs:element name="etel" maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="etel_neve" type="xs:string"/></xs:element>
      <xs:element name="teljes_ar" type="xs:int"/></xs:element>
    </xs:sequence>
    <xs:attribute name="Etel_ID" type="xs:int"/></xs:attribute>
    <xs:attribute name="ET_IDREF" type="xs:int"/></xs:attribute>
  </xs:complexType>
</xs:element>
<xs:element name="vevo" maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Nev" type="xs:string"/></xs:element>
      <xs:element name="Telefonszam" type="xs:int"/></xs:element>
      <xs:element name="Cim">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Utca" type="xs:string"/></xs:element>
            <xs:element name="Varos" type="xs:string"/></xs:element>
            <xs:element name="Hazszam" type="xs:int"/></xs:element>
            <xs:element name="Iranditoszam" type="xs:int"/></xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="vevo_ID" type="xs:int"/></xs:attribute>
  </xs:complexType>

```

```

    </xs:complexType>
  </xs:element>
  <xs:element name="bankartya" maxOccurs="unbounded">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Bank" type="xs:string"/></xs:element>
        <xs:element name="Lejarati_datum" type="xs:date"/></xs:element>
        <xs:element name="Tipus" type="xs:string"/></xs:element>
      </xs:sequence>
      <xs:attribute name="kartyaszam" type="xs:int"/></xs:attribute>
      <xs:attribute name="vevo_IDREF" type="xs:int"/></xs:attribute>
    </xs:complexType>
  </xs:element>
</xs:sequence>
<xs:attribute name="xmlns:xs" type="xs:string"/></xs:attribute>
<xs:attribute name="xs:noNamespaceSchemaLocation" type="xs:string"/></xs:attribute>
</xs:complexType>
</xs:element>
</xs:schema>

```

2.Feladat

2a/ Adatolvasás

```

package hu.dompars.ZW7DOR;

import java.io.File;
import java.io.IOException;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.w3c.dom.Text;
import org.xml.sax.SAXException;

public class DomReadZW7DOR {

    public static void main(String[] args) {

        File xmlFile = new File("XMLZW7DOR.xml");
        Document doc = introduceFile(xmlFile);

        if (doc != null) {
            doc.getDocumentElement().normalize();
            System.out.println("Root element: " + doc.getDocumentElement().getNodeName());
        } else {
            System.out.println("Document is null");
            System.exit(-1);
        }

        NodeList nodeList = doc.getDocumentElement().getChildNodes();
        String indent = "";
        listData(nodeList, indent);

    }

    public static Document introduceFile (File xmlFile){
        Document doc = null;

        try{
            DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
            DocumentBuilder dbBuilder = dbFactory.newDocumentBuilder();
            doc = dbBuilder.parse(xmlFile);
        } catch (ParserConfigurationException | SAXException | IOException e) {
            e.printStackTrace();
        }

        return doc;
    }
}

```

```

public static void listData(NodeList nodeList, String indent){
    indent += "\t";

    if(nodeList != null) {
        for (int i = 0; i < nodeList.getLength(); i++) {
            Node node = nodeList.item(i);
            if (node.getNodeType() == Node.ELEMENT_NODE && !node.getTextContent().trim().isEmpty()) {
                System.out.println(indent + "{" + node.getNodeName() + "}:");
                NodeList nodeList_new = node.getChildNodes();
                listData(nodeList_new, indent);
            } else if (node instanceof Text){
                String value = node.getNodeValue().trim();
                if (value.isEmpty()){
                    continue;
                }
                System.out.println(indent + node.getTextContent());
            }
        }
    }
}

```

Eredmények:

```

Root element: kiszallito_rendszer
  {etterem}:
    {Nyitvatartas}:
      11től 22:30ig
    {Nev}:
      Kalipszo
    {Elerhetoseg}:
      {Telefonszam}:
        209876543
      {Weboldal}:
        kalipszo.hu
  {etterem}:
    {Nyitvatartas}:
      9től 21ig
    {Nev}:
      Vendelin
    {Elerhetoseg}:
      {Telefonszam}:
        208765432
      {Weboldal}:
        vendelin.hu
  {etterem}:
    {Nyitvatartas}:
      8től 22ig
    {Nev}:
      Molnár Csárda
    {Elerhetoseg}:
      {Telefonszam}:
        306876546
      {Weboldal}:
        molnarcsarda.hu
  --

```

```
{futarok}:
  {futar}:
    {Nev}:
      Nagy Tíbor
    {Telefonszam}:
      202354342
  {futar}:
    {Nev}:
      Horváth Lajos
    {Telefonszam}:
      203521958
  {futar}:
    {Nev}:
      Palkó Patrik
    {Telefonszam}:
      202572553

{beszallito}:
  {Nev}:
    Kis Piac
  {Elerhetoseg}:
    kis.piac@gmail.com
  {Cim}:
    {Varos}:
      Kazincbarcika
    {Iranyitoszam}:
      3700
    {Utca}:
      Fő utca
    {Hazszam}:
      35

{beszallito}:
  {Nev}:
    Nagy piac
  {Elerhetoseg}:
    nagy.piac@gmail.com
  {Cim}:
    {Varos}:
      Encs
    {Iranyitoszam}:
      3896
    {Utca}:
      Fő utca
    {Hazszam}:
      25

{beszallito}:
  {Nev}:
    Közepes piac
  {Elerhetoseg}:
    kozepes.piac@gmail.com
  {Cim}:
    {Varos}:
      Miskolc
    {Iranyitoszam}:
      3885
    {Utca}:
      Fő utca
    {Hazszam}:
      15
```

```

{beszallitas}:
  {Hozzavalo}:
    banán
  {Datum}:
    2022.08.06
{beszallitas}:
  {Hozzavalo}:
    csoki
  {Datum}:
    2022.03.01
{beszallitas}:
  {Hozzavalo}:
    alma
  {Datum}:
    2022.10.21
-----
{etelek}:
  {etel}:
    {etel_neve}:
      Banános palacsinta
    {teljes_ar}:
      2500
  {etel}:
    {etel_neve}:
      Csokis zserbó
    {teljes_ar}:
      3400
  {etel}:
    {etel_neve}:
      Disznó almával a szájában
    {teljes_ar}:
      25000

```

```
{vevo}:
  {Nev}:
    Palkó Zsolti
  {Telefonszam}:
    202566311
  {Cim}:
    {Utca}:
      Károlyi Gáspár
    {Varos}:
      Szerencs
    {Hazzsam}:
      5
    {Iranditoszam}:
      3880
```

```
{vevo}:
  {Nev}:
    Lakatos Valentin
  {Telefonszam}:
    202599210
  {Cim}:
    {Utca}:
      Kálmán
    {Varos}:
      Budapest
    {Hazzsam}:
      31
    {Iranditoszam}:
      3550
```

```
{vevo}:
  {Nev}:
    Raul Patricio
  {Telefonszam}:
    202599901
  {Cim}:
    {Utca}:
      Nagy Lajos
    {Varos}:
      Pálháza
    {Hazzsam}:
      2
    {Iranditoszam}:
      3770
```

```
{bankartya}:  
  {Bank}:  
    OTP  
  {Lejarati_datum}:  
    2025.10.28  
  {Tipus}:  
    betéti kártya  
{bankartya}:  
  {Bank}:  
    MKB  
  {Lejarati_datum}:  
    2025.10.29  
  {Tipus}:  
    hitelkártya  
{bankartya}:  
  {Bank}:  
    CIB Bank  
  {Lejarati_datum}:  
    2025.10.30  
  {Tipus}:  
    többfunkciós kártya
```

2b/ Adatlekérdezés


```

package hu.domparse.ZW7DOR;

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

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import java.io.File;
import java.io.IOException;

public class DomQueryZW7DOR {

    public static void main(String[] args) throws SAXException, IOException, ParserConfigurationException {
        // TODO Auto-generated method stub

        //XML fájl meghívása
        File xmlFile = new File("XMLZW7DOR.xml");

        //Document builder definiálása
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = factory.newDocumentBuilder();

        //Fájl betöltése a documentum builderbe
        Document doc = dBuilder.parse(xmlFile);
        doc.getDocumentElement().normalize();

        //Futárok kilistázása

        System.out.println("");
        System.out.println("Futárok kilistazasa: ");
        System.out.println("");

        NodeList futarList = doc.getElementsByTagName("futar");
        for(int i=0;i<futarList.getLength();i++) {
            Node nNode = futarList.item(i);
            printFutar(nNode);
        }

        //Csokis zserbó étel kilistázása
        System.out.println("");
        System.out.println("Csokis zserbó étel kilistázása: ");
        System.out.println("");

        NodeList etellist = doc.getElementsByTagName("etel");
        for(int i=0;i<etellist.getLength();i++) {
            Node nNode = etellist.item(i);
            printFutar(nNode, "Csokis zserbó");
        }
    }
}

```

```

// 3400 feletti étel ki listázása

System.out.println("");
System.out.println("3400ft-os étel kilistazasa");
System.out.println("");

NodeList otezerList = doc.getElementsByTagName("étel");
for(int i=0;i<otezerList.getLength();i++) {
    Node nNode = otezerList.item(i);
    printOtezer(nNode, "3400");
}

System.out.println("");
System.out.println("Bankkártya kilistázása ");
//bankkkártya kilistázása másik megoldással

NodeList bankkartyaList =doc.getElementsByTagName("bankkartya");

for(int i=0;i<bankkartyaList.getLength();i++) {
    Node nNode = bankkartyaList.item(i);
    System.out.println("");
    printbankkartya(nNode);
}

//vevők kilistázása

System.out.println("");
System.out.println("vevők kilistázása");

NodeList vevoList = doc.getElementsByTagName("vevo");
for(int i=0;i<vevoList.getLength();i++) {
    Node nNode = vevoList.item(i);
    System.out.println("");
    printvevo(nNode);
}

}

//Futárok kilistázása
private static void printFutar(Node nNode) {
    if(nNode.getNodeType() == Node.ELEMENT_NODE) {
        Element elem = (Element) nNode;
        String F_ID = elem.getAttribute("F_ID");

        Node nNode1 = elem.getElementsByTagName("Név").item(0);
        String Nev = nNode1.getTextContent();

        Node nNode2 = elem.getElementsByTagName("Telefonszam").item(0);
        String Telefonszam = nNode2.getTextContent();
    }
}

```

```

        System.out.printf("F_ID: %s\n", F_ID);
        System.out.printf("Nev: %s\n", Nev);
        System.out.printf("Telefonszam %s\n", Telefonszam);

        System.out.println("");
    }
}

//Ételek ki listázása
private static void printFutar(Node nNode, String Nev) {
    if(nNode.getNodeType() == Node.ELEMENT_NODE) {
        Element elem = (Element) nNode;
        String F_ID = elem.getAttribute("Etel_ID");

        Node nNode1 = elem.getElementsByTagName("etel_neve").item(0);
        String Nev2 = nNode1.getTextContent();

        Node nNode2 = elem.getElementsByTagName("teljes_ar").item(0);
        String teljes_ar = nNode2.getTextContent();

        if(Nev2.equals(Nev)) {
            System.out.printf("F_ID: %s\n", F_ID);
            System.out.printf("Etel neve: %s\n", Nev2);
            System.out.printf("Teljes ar: %s\n", teljes_ar);

            System.out.println("");
        }
    }
}

// 3400ft-os ételek kilistázása
private static void printOtezer(Node nNode, String teljes_ar) {
    if(nNode.getNodeType() == Node.ELEMENT_NODE) {
        Element elem = (Element) nNode;
        String F_ID = elem.getAttribute("Etel_ID");

        Node nNode1 = elem.getElementsByTagName("etel_neve").item(0);
        String Felszereles = nNode1.getTextContent();

        Node nNode5 = elem.getElementsByTagName("teljes_ar").item(0);
        String teljes_ar2 = nNode5.getTextContent();

        if(teljes_ar2.equals(teljes_ar)) {
            System.out.printf("Etel_ID: %s\n", F_ID);
            System.out.printf("Etel neve: %s\n", Felszereles);

            System.out.printf("teljes ar %s\n", teljes_ar);

```

```

        System.out.println(" ");
    }
}

//bankkartyak kilistázása másik megoldással
private static void printbankkartya(Node nNode) {
    if(nNode.getNodeType()==Node.ELEMENT_NODE) {
        Element element =(Element) nNode;
        NodeList nList = element.getChildNodes();
        for (int j = 0; j < nList.getLength(); j++) {
            Node node2 = nList.item(j);
            if (node2.getNodeType() == Node.ELEMENT_NODE) {
                Element element2 = (Element) node2;
                System.out.println(node2.getNodeName()+" : "+node2.getTextContent());
            }
        }
    }
}

//vevok kilistázása
private static void printvevo(Node nNode) {
    if(nNode.getNodeType()==Node.ELEMENT_NODE) {
        Element element =(Element) nNode;
        NodeList nList = element.getChildNodes();
        for (int j = 0; j < nList.getLength(); j++) {
            Node node2 = nList.item(j);
            if (node2.getNodeType() == Node.ELEMENT_NODE) {
                Element element2 = (Element) node2;
                System.out.println(node2.getNodeName()+" : "+node2.getTextContent());
            }
        }
    }
}
}

```

és az eredmények :

Futárok kilistazasa:

F_ID: 1

Név: Nagy Tíbor

Telefonszam 202354342

F_ID: 2

Név: Horváth Lajos

Telefonszam 203521958

F_ID: 3

Név: Palkó Patrik

Telefonszam 202572553

Csokis zserbó étel kilistázása:

F_ID: 2

Etel neve: Csokis zserbó

Teljes ar: 3400

3400ft-os étel kilistazasa

Etel_ID: 2

Etel neve: Csokis zserbó

teljes_ar 3400

Bankkártya kilistázása

Bank : OTP

Lejarati_datum : 2025.10.28

Típus : betéti kártya

Bank : MKB

Lejarati_datum : 2025.10.29

Típus : hitelkártya

Bank : CIB Bank

Lejarati_datum : 2025.10.30

Típus : többfunkciós kártya

vevők kilistázása

Nev : Palkó Zsolti

Telefonszam : 202566311

Cím :

Károlyi Gáspár

Szerencs

5

3880

Nev : Lakatos Valentin

Telefonszam : 202599210

Cím :

Kálmán

Budapest

31

3550

Nev : Raul Patricio

Telefonszam : 202599901

Cím :

Nagy Lajos

Pálháza

2

3770

2a/ Adatmódosítás

```

package hu.domparse.ZW7DOR;

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

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.transform.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 DomModifyZW7DOR {

    public static void main(String[] args) throws SAXException, IOException, ParserConfigurationException, TransformerException {
        // TODO Auto-generated method stub

        //XML fájl meghívása
        File xmlFile = new File("XMLZW7DOR.xml");

        //Document builder definiálása
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = factory.newDocumentBuilder();

        //Fájl betöltése a DocumentBuilderbe
        Document doc = dBuilder.parse(xmlFile);
        doc.getDocumentElement().normalize();

        //A 1-es id-val rendelkező futar nevét változtatjuk
        NodeList nodes = doc.getElementsByTagName("futar");
        for(int i=0;i<nodes.getLength();i++) {
            Node node = nodes.item(i);
            if(node.getNodeType() == Node.ELEMENT_NODE) {
                if(node.getAttributes().getNamedItem("F_ID").getTextContent().equals("1")) {
                    NodeList childNodes = node.getChildNodes();
                    for(int j=0;j<childNodes.getLength();j++) {
                        Node childNode = childNodes.item(j);
                        if(childNode.getNodeName().equals("Nev")) {
                            childNode.setTextContent("Lajos Balázs");
                        }
                    }
                }
            }
        }
    }
}

```

```

//Új futárt adunk hozzá
Element futar = (Element)doc.getElementsByTagName("futarok").item(0);
futar.appendChild(createFutar(doc,"4","Katona János", "200000"));
//Új ételt adunk hozzá
Element etel = (Element)doc.getElementsByTagName("etelek").item(0);
etel.appendChild(createEtel(doc,"4", "Mákos Guba", "7000","4"));
//Módosítjuk a 1-es id-vel rendelkező beszállítót 4-es id-re
modifyId(doc, "beszallito", "Be_ID", "1", "4");
//Módosítjuk a 1-es id-vel rendelkező éttermet 4-es id-re
modifyId(doc, "etterem", "ET_ID", "1", "4");

//Lementjük a módosított dokumentumot
SaveAsDoc(doc, "teszt.xml");
}

//Lementjük a módosított xml dokumentumot
public static void SaveAsDoc(Document doc, String filename) throws TransformerException {
    TransformerFactory transformerFactory = TransformerFactory.newInstance();
    Transformer transf = transformerFactory.newTransformer();

    transf.setOutputProperty(OutputKeys.ENCODING, "UTF-8");
    transf.setOutputProperty(OutputKeys.INDENT, "yes");
    transf.setOutputProperty("{https://xml.apache.org/xslt}indent-amount", "2");

    DOMSource source = new DOMSource(doc);

    File myFile = new File(filename);

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

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

//futárt hozzunk létre
private static Node createFutar(Document doc, String F_ID, String nev, String Telefonszam) {

    Element user = doc.createElement("futar");

    user.setAttribute("F_ID", F_ID);
    user.appendChild(createElement(doc, "Nev", nev));
    user.appendChild(createElement(doc, "Telefonszam", Telefonszam));

    return user;
}

```



```

//ételt hozzunk létre
private static Node createEtel(Document doc, String Etel_ID, String etel_neve, String teljes_ar, String ET_IDREF) {

    Element user = doc.createElement("etel");

    user.setAttribute("Etel_ID", Etel_ID);
    user.appendChild(createElement(doc, "etel_neve", etel_neve));
    user.appendChild(createElement(doc, "teljes_ar", teljes_ar));
    user.appendChild(createElement(doc, "ET_IDREF", ET_IDREF));

    return user;
}

//Elementet hozzunk létre
private static Node createElement(Document doc, String name, String value) {

    Element node = doc.createElement(name);
    node.appendChild(doc.createTextNode(value));

    return node;
}

//Módosítjuk az id-t
public static void modifyId(Document doc, String tagName, String attrName, String oldData, String newData) {

    NodeList nodes = doc.getElementsByTagName(tagName);
    for (int i = 0; i < nodes.getLength(); i++) {
        Node node = nodes.item(i);
        if (node.getNodeType() == Node.ELEMENT_NODE) {
            if (node.getAttribute().getNamedItem(attrName).getTextContent().equals(oldData)) {
                node.getAttribute().getNamedItem(attrName).setTextContent(newData);
            }
        }
    }
}

```

Eredmények :

```
<futar ET_IDREF="1" F_ID="1">
  <Nev>Lajos Balázs</Nev>
  <Telefonszam>202354342</Telefonszam>
</futar>

<futar ET_IDREF="2" F_ID="2">
  <Nev>Horváth Lajos</Nev>
  <Telefonszam>203521958</Telefonszam>
</futar>

<futar ET_IDREF="3" F_ID="3">
  <Nev>Palkó Patrik</Nev>
  <Telefonszam>202572553</Telefonszam>
</futar>

<futar ET_IDREF="4" F_ID="4">
  <Nev>Katona János</Nev>
  <Telefonszam>200000</Telefonszam>
</futar>
```

```
<etel ET_IDREF="1" Etel_ID="1">
  <etel_neve>Banános palacsinta</etel_neve>
  <teljes_ar>2500</teljes_ar>
</etel>

<etel ET_IDREF="2" Etel_ID="2">
  <etel_neve>Csokis zserbó</etel_neve>
  <teljes_ar>3400</teljes_ar>
</etel>

<etel ET_IDREF="3" Etel_ID="3">
  <etel_neve>Disznó almával a szájában</etel_neve>
  <teljes_ar>25000</teljes_ar>
</etel>

<etel ET_IDREF="4" Etel_ID="4">
  <etel_neve>Mákos Guba</etel_neve>
  <teljes_ar>7000</teljes_ar>
</etel>
```

```
<beszallito Be_ID="4">
  <Nev>Kis Piac</Nev>
  <Elerhetoseg>kis.piac@gmail.com</Elerhetoseg>
  <Cim>
    <Varos>Kazincbarcika</Varos>
    <Iranyitoszam>3700</Iranyitoszam>
    <Utca>Fő utca</Utca>
    <Hazszam>35</Hazszam>
  </Cim>
</beszallito>
```

```
<etterem ET_ID="4">  
  
  <Nyitvatartas>11től 22:30ig</Nyitvatartas>  
  
  <Nev>Kalipszo</Nev>  
  
  <Elerhetoseg>  
    <Telefonszam>209876543</Telefonszam>  
  
    <Weboldal>kalipszo.hu</Weboldal>  
  </Elerhetoseg>  
</etterem>
```