

# JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Pizzázók működése

Készítette: **Zsiros Nikolett**

Neptun kód: **MOIZHL**

## **A feladat leírása:**

A beadandó témája egy olyan adatbázis, amely több pizzázót kezel. Rákereshetünk benne a pizzázóban dolgozó futárokra, vagy beszállítókra, a vevő adatait is lekérdezhethetjük.

Összesen 5 egyed található meg benne:

Ezek a Beszállító, Pizzázó, Pizza, Vevő, Futár.

## **Egyedek közötti kapcsolat:**

### **◆ Pizzázó és Futár:**

A Pizzázó és a Futár egyedek között egy a többhöz kapcsolat van, mivel egy pizzázó alkalmazhat több futárt, de egy futár csak egy pizzázónál dolgozik.

### **◆ Pizzázó és Beszállító:**

A Pizzázó és a Beszállító egyedek között több a többhöz kapcsolat van, mivel egy pizzázó rendelhet több beszállítótól, valamint egy beszállító beszállíthat több pizzázónak is. A kapcsolat paraméterei: a Hozzávalók, amely a beszállító által beszállított hozzávalókat jelenti, valamint a Dátum, azaz a beszállítás dátuma.

### **◆ Pizzázó és Pizza:**

A Pizzázó és a Pizza egyedek között egy a többhöz kapcsolat van, mivel egy pizzázónak lehet több pizzája, de egy pizza csak egy pizzázóhoz tartozhat.

### **◆ Pizza és Vevő:**

A Pizza és a Vevő egyedek között több a többhöz kapcsolat van, mivel egy vevő rendelhet többfajta pizzát, és egy pizzát több vevő is megvehet.

## **Az ER modell egyedei és tulajdonságai:**

### **◆ A Vevő egyed tulajdonságai**

- VevőID: A Vevő egyed elsődleges kulcsa.
- Név: A vevő neve.
- Telefonszám: A vevő telefonszáma.
- Cím: Összetett tulajdonság. A vevő címe.

### **◆ A Pizza egyed tulajdonságai**

- PizzaID: A Pizza egyed elsődleges kulcsa.
- Teljes\_ár: A rendelt pizza/pizzák teljes ára. Származtatott tulajdonság.
- Pizza\_neve: A pizza neve.
- Méret: Többértékű tulajdonság. A pizza méretét tárolja.

### **◆ A Futár egyed tulajdonságai**

- FutárID: A Futár egyed elsődleges kulcsa.
- Telefonszám: A futár telefonszáma.
- Név: A futár neve.

### ◆ A Beszállító egyed tulajdonságai

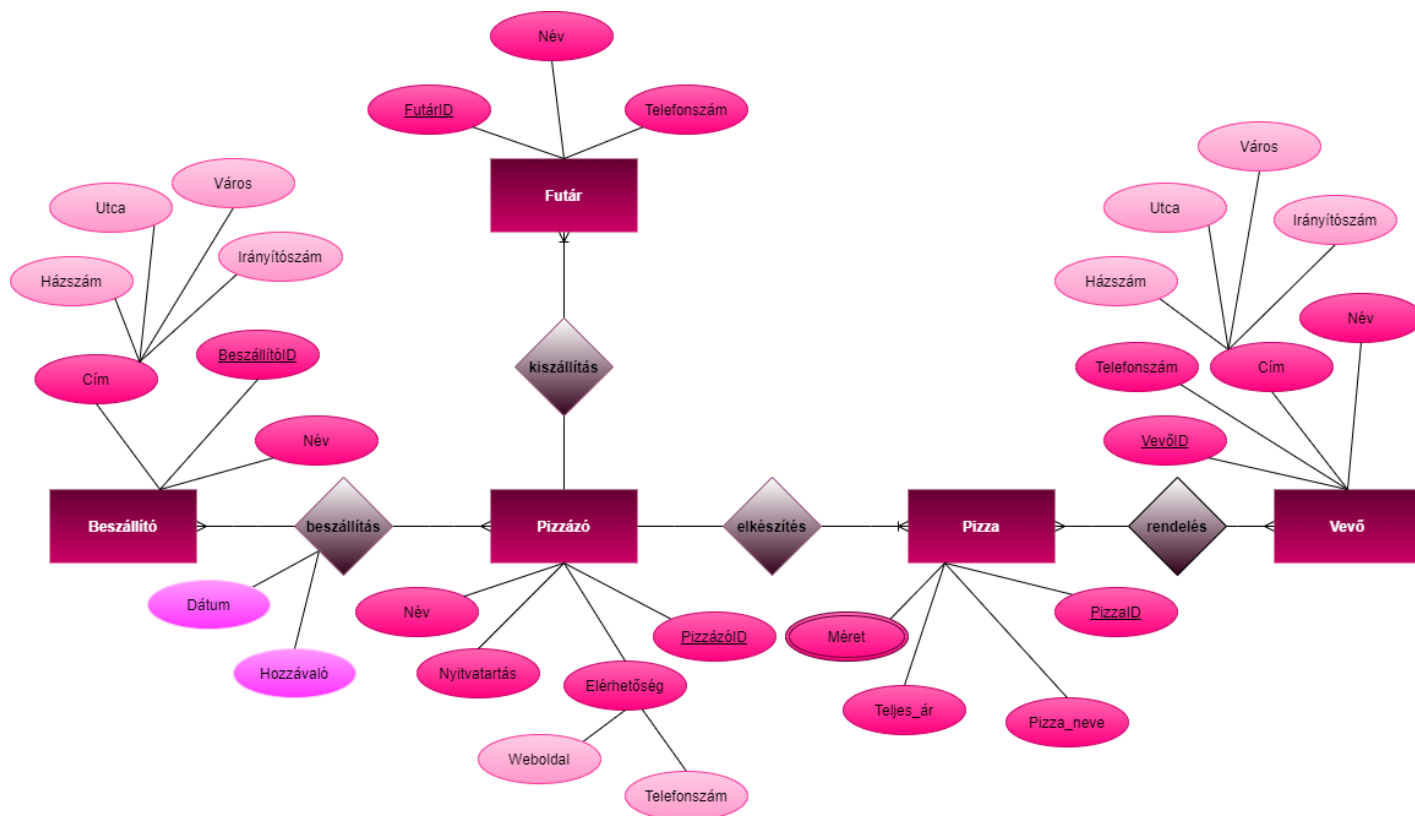
- BeszállítóID: A Beszállító egyed elsődleges kulcsa.
- Név: A beszállító cég neve.
- Cím: Összetett tulajdonság. A beszállító cég címe.

### ◆ A Pizzázó egyed tulajdonságai

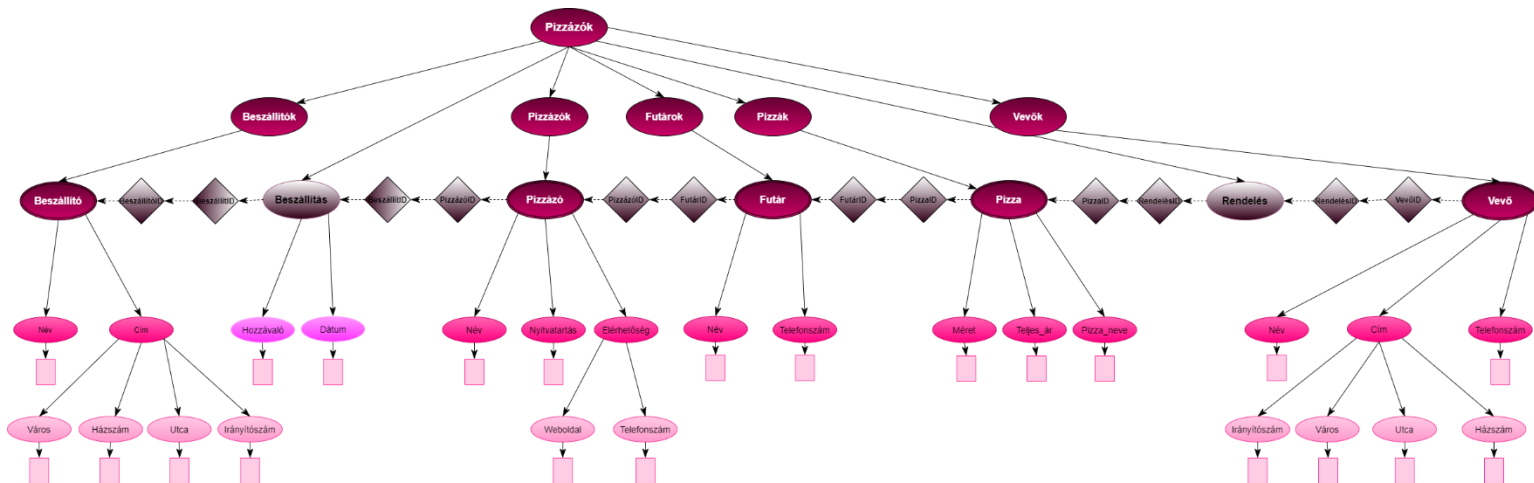
- PizzázóID: A Pizzázó egyed elsődleges kulcsa.
- Nyitva tartás: A pizzázó nyitva tartási ideje.
- Név: A pizzázó neve.
- Elérhetőség: Összetett tulajdonság. A pizzázó elérhetőségei.

## 1.feladat

### a) Az adatbázis ER modellje:



**b) Az adatbázis konvertálása XDM modellre:**



**c) Az XDM modell alapján XML dokumentum készítése:**

```

<?xml version="1.0" encoding="utf-8" standalone="no"?><Gyorsetterem
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="XMLSchemaMoizhl.xsd">
  <Pizzazok>
    <Pizzazo PizzazoID="1">
      <Nev>Pizza Hut</Nev>
      <Nyitvatartas>08-22</Nyitvatartas>
      <Elerhetoseg>
        <Weboldal>pizza.hut.hu</Weboldal>
        <Telefonszam>(20)9876543</Telefonszam>
      </Elerhetoseg>
    </Pizzazo>
    <Pizzazo PizzazoID="2">
      <Nev>Pizza Bazis</Nev>
      <Nyitvatartas>10-19</Nyitvatartas>
      <Elerhetoseg>
        <Weboldal>pizza.bazis.hu</Weboldal>
        <Telefonszam>(20)8765432</Telefonszam>
      </Elerhetoseg>
    </Pizzazo>
  </Pizzazok>
  <Futarak>
    <Futar FutarID="1" PizzazoID="1">
      <Nev>Kiss József</Nev>
      <Telefonszam>(20)5234534</Telefonszam>
    </Futar>
    <Futar FutarID="2" PizzazoID="2">
      <Nev>Nagy János</Nev>
      <Telefonszam>(30)3452345</Telefonszam>
    </Futar>
  </Futarak>
  <Beszallitas BeszallitID="1" BeszallitoID="1" PizzazoID="1">
    <Hozzavalo>Majonéz</Hozzavalo>
    <Datum>2021-11-29</Datum>
  </Beszallitas>
  <Beszallitas BeszallitID="2" BeszallitoID="2" PizzazoID="2">
    <Hozzavalo>Ketchup</Hozzavalo>
    <Datum>2021-10-29</Datum>
  </Beszallitas>
</Gyorsetterem>
  
```

```

<Beszallitas BeszallitID="2" BeszallitoID="2" PizzazoID="2">
  <Hozzavalo>Ketchup</Hozzavalo>
  <Datum>2021-10-29</Datum>
</Beszallitas>

<Beszallitok>
  <Beszallito BeszallitoID="1">
    <Nev>Városi Piac</Nev>
    <Cim>
      <Iranyitoszam>3509</Iranyitoszam>
      <Varos>Miskolc</Varos>
      <Utca>Fő utca</Utca>
      <Hatszam>13</Hatszam>
    </Cim>
  </Beszallito>
  <Beszallito BeszallitoID="2">
    <Nev>Kis Zöldséges</Nev>
    <Cim>
      <Iranyitoszam>1032</Iranyitoszam>
      <Varos>Budapest</Varos>
      <Utca>Arany János út</Utca>
      <Hatszam>5</Hatszam>
    </Cim>
  </Beszallito>
</Beszallitok>

<Pizzak>
  <Pizza PizzaID="1" PizzazoID="1">
    <Meret>40</Meret>
    <Teljes_ar>4000</Teljes_ar>
    <Pizza_neve>Piedone</Pizza_neve>
  </Pizza>
  <Pizza PizzaID="2" PizzazoID="2">
    <Meret>40</Meret>
    <Teljes_ar>3000</Teljes_ar>
    <Pizza_neve>Tengergyumolcse</Pizza_neve>
  </Pizza>
  <Pizza PizzaID="3" PizzazoID="2">
    <Meret>30</Meret>
    <Teljes_ar>4300</Teljes_ar>
    <Pizza_neve>Pepperoni</Pizza_neve>
  </Pizza>
</Pizzak>

<Rendeles PizzaID="1" RendelesID="1" VevoID="1"/>
<Rendeles PizzaID="2" RendelesID="2" VevoID="2"/>

<Vevok>
  <Vevo VevoID="1">
    <Nev>Kiss János</Nev>
    <Cim>
      <Iranyitoszam>3509</Iranyitoszam>
      <Varos>Miskolc</Varos>
      <Utca>Virág utca</Utca>
      <Hatszam>1</Hatszam>
    </Cim>
    <Telefonszam>(70)4343433</Telefonszam>
  </Vevo>

```

```

        <Vevo VevoID="2">
            <Nev>Nagy Géza</Nev>
            <Cim>
                <Iranyitoszam>3510</Iranyitoszam>
                <Varos>Miskolc</Varos>
                <Utca>Szeder utca</Utca>
                <Hazszam>32</Hazszam>
            </Cim>
            <Telefonszam>(30)1212123</Telefonszam>
        </Vevo>
    </Vevok>

</Pizzazok>

</Gyorsetterem>

```

**d) Az XML dokumentum alapján XML Schema készítése:**

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" attributeFormDefault="unqualified"
    elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="Gyorsetterem">
        <xs:complexType>
            <xs:sequence>
                <xs:choice maxOccurs="unbounded">
                    <xs:element name="Beszallitok" />
                    <xs:element name="Beszallito" type="BeszallitoType" />
                    <xs:element name="Pizzazok" />
                    <xs:element name="Pizzazo" type="PizzazoType" />
                    <xs:element name="Beszallitas" type="BeszallitasType" />
                    <xs:element name="Futarok" />
                    <xs:element name="Futar" type="FutarType" />
                    <xs:element name="Pizzak" />
                    <xs:element name="Pizza" type="PizzaType" />
                    <xs:element name="Rendeles" type="RendelesType" />
                    <xs:element name="Vevok" />
                    <xs:element name="Vevo" type="VevoType" />
                </xs:choice>
            </xs:sequence>
        </xs:complexType>

        <xs:key name="BeszallitoID_PK">
            <xs:selector xpath="Beszallito"/>
            <xs:field xpath="@BeszallitoID"/>
        </xs:key>

        <xs:key name="PizzazoID_PK">
            <xs:selector xpath="Pizzazo"/>
            <xs:field xpath="@PizzazoID"/>
        </xs:key>

```

```

        <xs:key name="PizzaID_PK">
            <xs:selector xpath="Pizza"/>
            <xs:field xpath="@PizzaID"/>
        </xs:key>

        <xs:key name="VevoID_PK">
            <xs:selector xpath="Vevo"/>
            <xs:field xpath="@VevoID"/>
        </xs:key>

        <xs:key name="FutarID_PK">
            <xs:selector xpath="Futar"/>
            <xs:field xpath="@FutarID"/>
        </xs:key>

```

```

    <xs:key name="BeszallitID_PK">
      <xs:selector xpath="Beszallitas"/>
      <xs:field xpath="@BeszallitID"/>
    </xs:key>

    <xs:key name="RendelesID_PK">
      <xs:selector xpath="Rendeles"/>
      <xs:field xpath="@RendelesID"/>
    </xs:key>

    <xs:keyref name="VevoID_FK" refer="VevoID_PK">
      <xs:selector xpath="Rendeles"/>
      <xs:field xpath="@VevoID"/>
    </xs:keyref>

    <xs:keyref name="PizzazoID_FK" refer="PizzazoID_PK">
      <xs:selector xpath="Futar/Pizza/Beszallitas"/>
      <xs:field xpath="@PizzazoID"/>
    </xs:keyref>

    <xs:keyref name="PizzaID_FK" refer="PizzaID_PK">
      <xs:selector xpath="Rendeles"/>
      <xs:field xpath="@PizzaID"/>
    </xs:keyref>

    <xs:keyref name="BeszallitoID_FK" refer="BeszallitoID_PK">
      <xs:selector xpath="Beszallitas"/>
      <xs:field xpath="@BeszallitoID"/>
    </xs:keyref>

  </xs:element>

  <xs:complexType name="BeszallitoType">
    <xs:sequence>
      <xs:element name="Nev" type="xs:string" />
      <xs:element name="Cim" type="CimType" />
    </xs:sequence>
    <xs:attribute name="BeszallitoID" type="xs:string" use="required" />
  </xs:complexType >

  <xs:complexType name="CimType">
    <xs:sequence>
      <xs:element name="Iranyitoszam" type="xs:unsignedShort" />
      <xs:element name="Varos" type="xs:string" />
      <xs:element name="Utca" type="xs:string" />
      <xs:element name="Hazszam" type="xs:unsignedByte" />
    </xs:sequence>
  </xs:complexType >

  <xs:complexType name="PizzazoType">
    <xs:sequence>
      <xs:element name="Nev" type="xs:string" />
      <xs:element name="Nyitvatartas" type="NyitvatartasType" />
      <xs:element name="Elerhetoseg" type="ElerhetosegType" />
    </xs:sequence>
    <xs:attribute name="PizzazoID" type="xs:unsignedByte" use="required" />
  </xs:complexType >

```

```

<xs:complexType name="ElerhetosegType">
  <xs:sequence>
    <xs:element name="Weboldal" type="xs:string" />
    <xs:element name="Telefonszam" type="TelefonszamType" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="BeszallitasType">
  <xs:sequence>
    <xs:element name="Hozzavalo" type="xs:string" />
    <xs:element name="Datum" type="xs:date" />
  </xs:sequence>
  <xs:attribute name="BeszallitID" type="xs:unsignedByte" use="required" />
  <xs:attribute name="BeszallitoID" type="xs:unsignedByte" use="required" />
  <xs:attribute name="Pizzazoid" type="xs:unsignedByte" use="required" />
</xs:complexType>

<xs:complexType name="FutarType">
  <xs:sequence>
    <xs:element name="Nev" type="xs:string" />
    <xs:element name="Telefonszam" type="TelefonszamType" />
  </xs:sequence>
  <xs:attribute name="FutarID" type="xs:unsignedByte" use="required" />
  <xs:attribute name="Pizzazoid" type="xs:unsignedByte" use="required" />
</xs:complexType>

<xs:complexType name="PizzaType">
  <xs:sequence>
    <xs:element name="Meret" type="xs:unsignedByte" />
    <xs:element name="Teljes_ar" type="xs:unsignedShort" />
    <xs:element name="Pizza_neve" type="xs:string" />
  </xs:sequence>
  <xs:attribute name="PizzaID" type="xs:unsignedByte" use="required" />
  <xs:attribute name="Pizzazoid" type="xs:unsignedByte" use="required" />
</xs:complexType>

```

```

<xs:complexType name="Rendelestyp">
  <xs:attribute name="PizzaID" type="xs:unsignedByte" use="required" />
  <xs:attribute name="RendelestID" type="xs:unsignedByte" use="required" />
  <xs:attribute name="VevoID" type="xs:unsignedByte" use="required" />
</xs:complexType>

<xs:complexType name="VevoType">
  <xs:sequence>
    <xs:element name="Nev" type="xs:string" />
    <xs:element name="Cim" type="CimType" />
    <xs:element name="Telefonszam" type="TelefonszamType" />
  </xs:sequence>
  <xs:attribute name="VevoID" type="xs:unsignedByte" use="required" />
</xs:complexType>

```



```

<xs:simpleType name="NyitvatartasType">
  <xs:restriction base="xs:string">
    <xs:pattern value="[0-9][0-9]-[0-9][0-9]" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="TelefonszamType">
  <xs:restriction base="xs:string">
    <xs:pattern value="\([0-9]{2}\)[0-9]{7}" />
  </xs:restriction>
</xs:simpleType>

</xs:schema>

```

## 2.feladat

### a) DOM JAVA olvasás kommenttel

```

1 package hu.domparse.moizhl;
2
3 import org.w3c.dom.Document;
4 import org.w3c.dom.Element;
5 import org.w3c.dom.Node;
6 import org.w3c.dom.NodeList;
7 import org.xml.sax.SAXException;
8 import javax.xml.parsers.DocumentBuilder;
9 import javax.xml.parsers.DocumentBuilderFactory;
10 import javax.xml.parsers.ParserConfigurationException;
11 import java.io.File;
12 import java.io.IOException;
13
14 public class DomReadMOIZHL {
15
16     public static void main(String[] args) {
17         try {
18             File xmlFile = new File("XMLmoizhl.xml"); // az XML fájl beolvasása
19             DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance(); //XML doksiból DOM doksi létrehozása
20             DocumentBuilder dBuilder = factory.newDocumentBuilder();
21             Document doc = dBuilder.parse(xmlFile); // a dokumentum lekérése
22             doc.getDocumentElement().normalize();
23             System.out.println("Pizzázó adatok lekérése:");
24             Read(doc); //A programunk fő metodusa, itt hívodik meg Read fgv
25
26             //Kivételkezelés
27         } catch (ParserConfigurationException pce) {
28             pce.printStackTrace();
29         } catch (IOException ioe) {
30             ioe.printStackTrace();
31         } catch (SAXException sae) {
32             sae.printStackTrace();
33         }
34     }
35 }
36
37

```

```

37
38 public static void Read(Document doc) {
39     NodeList nList = doc.getElementsByTagName("Pizzazo"); // Pizzazo taggal rendelkezo elemek lekereasa listaba
40
41     for (int i = 0; i < nList.getLength(); i++) { // listan vegigmegeyunk
42         Node nNode = nList.item(i); // lekeriuk a lista aktualis elemet, Elementte konvertaljuk
43         Element element = (Element) nNode;
44         // lekeriuk az attributumokat, majd azok segitsegevel meghivjuk a definialt metodusokat
45
46         if (nNode.getNodeType() == Node.ELEMENT_NODE) {
47             String Nev = element.getElementsByTagName("Nev").item(0).getTextContent(); // darabszam lekerezesse
48             String Nyitvatartas = element.getElementsByTagName("Nyitvatartas").item(0).getTextContent();
49             String Weboldal = element.getElementsByTagName("Weboldal").item(0).getTextContent();
50             String Telefonszam = element.getElementsByTagName("Telefonszam").item(0).getTextContent();
51
52             String BeszallitasID = element.getAttribute("BeszallitasID");
53             String FutarID = element.getAttribute("FutarID");
54
55             String PizzaID = element.getAttribute("PizzaID");
56
57             System.out.println("\n-----" + (i + 1)
58                 + ". Pizzazo-----");
59             System.out.println("Pizzazo adatok: \n\tNév:\t" + Nev + "\n\tNyitvatartas:\t" + Nyitvatartas
60                 + "\n\tWeboldal:\t" + Weboldal + "\n\tTelefonszám:\t" + Telefonszam);
61
62             ReadBeszallitasById(doc, BeszallitasID);
63             ReadFutarById(doc, FutarID);
64             ReadPizzaById(doc, PizzaID);
65
66         }
67     }
68 }
69
70 }
71 // fa struktúra miatt az attribútumban megadott id alapján kérdezzük le az egyes rendeléshez tartozó elemeket

```

```

71 // fa struktúra miatt az attribútumban megadott id alapján kérdezzük le az egyes rendeléshez tartozó elemeket
72 // A leatobb objektum rendelkerik leszammarozattal, amelyet esz kiabb metódus kender le, az attribútumban megadott ID alapján
73 public static void ReadFutarById(Document doc, String FutarID) {
74     NodeList nList = doc.getElementsByTagName("Futarok");
75     for (int i = 0; i < nList.getLength(); i++) {
76         Node nNode = nList.item(i);
77         Element element = (Element) nNode;
78         if (nNode.getNodeType() == Node.ELEMENT_NODE) {
79             if (element.getAttribute("FutarID").equals(FutarID)) {
80                 String Nev = element.getElementsByTagName("Nev").item(0).getTextContent();
81                 String Telefonszam = element.getElementsByTagName("Telefonszam").item(0).getTextContent();
82                 System.out.println("Futár adatok: \n\tNév:\t" + Nev + "\n\tTelefonszam:\t" + Telefonszam); //Konzolra kiiras
83             }
84         }
85     }
86 }
87
88 public static void ReadBeszallitasById(Document doc, String BeszallitasID) {
89     NodeList nList = doc.getElementsByTagName("Beszallitas");
90     for (int i = 0; i < nList.getLength(); i++) {
91         Node nNode = nList.item(i);
92         Element element = (Element) nNode;
93         if (nNode.getNodeType() == Node.ELEMENT_NODE) {
94             if (element.getAttribute("BeszallitasID").equals(BeszallitasID)) {
95                 String Datum = element.getElementsByTagName("Datum").item(0).getTextContent();
96                 String Hozzavaló = element.getElementsByTagName("Hozzavaló").item(0).getTextContent();
97                 System.out.println("Beszállítás adatok: \n\tDátum\t" + Datum + "\n\tHozzávaló:\t" + Hozzavaló); //Konzolra kiiras
98                 String BeszallitoID = element.getAttribute("BeszallitoID");
99
100                 ReadBeszallitoById(doc, BeszallitoID);
101             }
102         }
103     }
104 }
105 }
106
107 }

```

```

107 public static void ReadBeszallitoById(Document doc, String BeszallitoID) {
108     NodeList nList = doc.getElementsByTagName("Beszallito");
109     for (int i = 0; i < nList.getLength(); i++) {
110         Node nNode = nList.item(i);
111         Element element = (Element) nNode;
112         if (nNode.getNodeType() == Node.ELEMENT_NODE) {
113             if (element.getAttribute("BeszallitoID").equals(BeszallitoID)) {
114                 String Nev = element.getElementsByTagName("Nev").item(0).getTextContent();
115                 String Iranyitoszam = element.getElementsByTagName("Iranyitoszam").item(0).getTextContent();
116                 String Varos = element.getElementsByTagName("Varos").item(0).getTextContent();
117                 String Utca = element.getElementsByTagName("Utca").item(0).getTextContent();
118                 String Hazszam = element.getElementsByTagName("Hazszam").item(0).getTextContent();
119
120                 System.out.println("Besz ll t  adatok: \n\tN v:\t" + Nev + "\n\tIr ny t sz m:\t" + Iranyitoszam
121                     + "\n\tV ros:\t" + Varos + "\n\tUtca:\t" + Utca + "\n\tH zsz m:\t" + Hazszam); //konzolra kiiras
122             }
123         }
124     }
125 }
126 public static void ReadPizzaById(Document doc, String PizzaID) {
127     NodeList nList = doc.getElementsByTagName("Pizzak");
128     for (int i = 0; i < nList.getLength(); i++) {
129         Node nNode = nList.item(i);
130         Element element = (Element) nNode;
131         if (nNode.getNodeType() == Node.ELEMENT_NODE) {
132             if (element.getAttribute("PizzaID").equals(PizzaID)) {
133                 String Meret = element.getElementsByTagName("Meret").item(0).getTextContent();
134                 String Teljes_ar = element.getElementsByTagName("Teljes_ar").item(0).getTextContent();
135                 String Pizza_neve = element.getElementsByTagName("Pizza_neve").item(0).getTextContent();
136                 System.out.println("Pizza adatok: \n\tMeret:\t" + Meret + "\n\tTeljes_ar:\t" + Teljes_ar
137                     + "\n\tPizza_neve:\t" + Pizza_neve); //konzolra kiiras
138                 String RendelesID = element.getAttribute("RendelesID");
139                 ReadRendelesbyId(doc, RendelesID);
140             }
141         }
142     }
143 }
144

```

```

145 public static void ReadRendelesbyId(Document doc, String RendelesID) {
146     NodeList nList = doc.getElementsByTagName("Rendeles");
147     for (int i = 0; i < nList.getLength(); i++) {
148         Node nNode = nList.item(i);
149         Element element = (Element) nNode;
150         if (nNode.getNodeType() == Node.ELEMENT_NODE) {
151             if (element.getAttribute("RendelesID").equals(RendelesID)) {
152
153                 String VevoID = element.getAttribute("VevoID");
154                 ReadVevoById(doc, VevoID);
155
156                 String PizzaID = element.getAttribute("PizzaID");
157                 ReadPizzaById(doc, PizzaID);
158             }
159         }
160     }
161 }
162 public static void ReadVevoById(Document doc, String VevoID) {
163     NodeList nList = doc.getElementsByTagName("Vevok");
164     for (int i = 0; i < nList.getLength(); i++) {
165         Node nNode = nList.item(i);
166         Element element = (Element) nNode;
167         if (nNode.getNodeType() == Node.ELEMENT_NODE) {
168             if (element.getAttribute("VevoID").equals(VevoID)) {
169                 String Nev = element.getElementsByTagName("Nev").item(0).getTextContent();
170                 String Telefonszam = element.getElementsByTagName("Telefonszam").item(0).getTextContent();
171                 String Iranyitoszam = element.getElementsByTagName("Iranyitoszam").item(0).getTextContent();
172                 String Varos = element.getElementsByTagName("Varos").item(0).getTextContent();
173                 String Utca = element.getElementsByTagName("Utca").item(0).getTextContent();
174                 String Hazszam = element.getElementsByTagName("Hazszam").item(0).getTextContent();
175                 System.out.println("Vev  adatok: \n\tN v:\t" + Nev + "\n\tTelefonszam:\t" + Telefonszam +
176                     "\n\tIr ny t sz m:\t" + Iranyitoszam + "\n\tV ros:\t" + Varos + "\n\tUtca:\t" + Utca +
177                     "\n\tH zsz m:\t" + Hazszam); //konzolra kiiras
178             }
179         }
180     }
181 }
182

```

## b) DOM JAVA lekérdezés kommenttel

```
1 package hu.domparse.moizhl;
2
3 import java.io.File;
4 import java.io.IOException;
5 import java.util.Scanner;
6
7 import javax.xml.parsers.DocumentBuilder;
8 import javax.xml.parsers.DocumentBuilderFactory;
9 import javax.xml.parsers.ParserConfigurationException;
10 import javax.xml.transform.TransformerException;
11
12 import org.w3c.dom.Document;
13 import org.w3c.dom.Element;
14 import org.w3c.dom.Node;
15 import org.w3c.dom.NodeList;
16 import org.xml.sax.SAXException;
17
18
19 public class DomQueryMOIZHL {
20
21     public static void main(String[] args)
22         throws ParserConfigurationException, IOException, SAXException, TransformerException {
23
24         File xmlFile = new File("XMLmoizhl.xml"); // xml fájl beolvasása
25         DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance(); // olvasás lehetővé tetele, /XML doksiból DOM doksi létrehozása
26         DocumentBuilder dBuilder = factory.newDocumentBuilder();
27         Document doc = dBuilder.parse(xmlFile);
28         doc.getDocumentElement().normalize();
29
30         System.out.println("Root elem: " + doc.getDocumentElement().getNodeName());
31         System.out.println("-----");
32         LoadPizzazoQuery(doc);
33
34
35     public static void LoadPizzazoQuery(Document doc) throws TransformerException {
36         NodeList nodeList = doc.getElementsByTagName("Pizzazo"); // Pizzazo elemek listázása
37         String Pizzazo;
38         Element element = null;
39         Node nNode = null;
40         for (int i = 0; i < nodeList.getLength(); i++) {
41             nNode = nodeList.item(i);
42             element = (Element) nNode;
43             String Nev = element.getElementsByTagName("Nev").item(0).getTextContent();
44             System.out.println((i + 1) + " " + Nev);
45         }
46         // Pizzazo választás
47         System.out.println("Írja be annak a Pizzazonak a nevét, amelyiknek latni szeretne a Pizzak adatait:");
48         Scanner sc = new Scanner(System.in);
49         Pizzazo = sc.nextLine();
50         for (int i = 0; i < nodeList.getLength(); i++) {
51             nNode = nodeList.item(i);
52             element = (Element) nNode;
53             if (nNode.getNodeType() == Node.ELEMENT_NODE) {
54
55                 if (Pizzazo.equals("Pizza Hut")) {
56                     LoadPizzaQuery(doc, "1");
57                     break;
58                 }
59
60                 if (Pizzazo.equals("Pizza Bazis")) {
61                     LoadPizzaQuery(doc, "2");
62                     break;
63                 }
64             }
65         }
66     }
67     sc.close();
68 }
69 }
```

```

70 //A kiválasztott Pizzazo Pizzainak kiírása
71 public static void LoadPizzaQuery(Document doc, String id) throws TransformerException {
72     NodeList nodeList = doc.getElementsByTagName("Pizza");
73     int Pizza = 0;
74
75     for (int i = 0; i < nodeList.getLength(); i++) {
76         Node nNode = nodeList.item(i);
77         Element element = (Element) nNode;
78         String PizzazoID = element.getAttribute("PizzazoID");
79
80         if (nNode.getNodeType() == Node.ELEMENT_NODE) {
81
82             if (id.equals(PizzazoID)) {
83                 Pizza+= 1;
84
85                 System.out.println(Pizza + ". Pizza adatai:");
86                 String PizzaID = element.getAttribute("id");
87                 DomReadMOIZHL.ReadPizzaById(doc, PizzaID);
88
89             }
90         }
91     }
92 }
93 }
94
95

```

c) DOM JAVA módosítás kommenttel

```

1 package hu.domparse.moizhl;
2
3 import org.w3c.dom.Document;
4 import org.w3c.dom.Element;
5 import org.w3c.dom.Node;
6 import org.w3c.dom.NodeList;
7 import org.xml.sax.SAXException;
8 import javax.xml.parsers.DocumentBuilder;
9 import javax.xml.parsers.DocumentBuilderFactory;
10 import javax.xml.parsers.ParserConfigurationException;
11 import javax.xml.transform.Transformer;
12 import javax.xml.transform.TransformerException;
13 import javax.xml.transform.TransformerFactory;
14 import javax.xml.transform.dom.DOMSource;
15 import javax.xml.transform.stream.StreamResult;
16 import java.io.File;
17 import java.io.IOException;
18 import java.util.Scanner;
19
20 public class DomModifyMOIZHL {
21     public static void main(String[] args)
22         throws ParserConfigurationException, IOException, SAXException, TransformerException {
23         File xmlFile = new File("XMLmoizhl.xml"); // xml fájl bekérése
24         DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance(); // az olvasást lehetővé tesszük
25         DocumentBuilder dBuilder = factory.newDocumentBuilder(); //XML doksból DOM doksi létrehozása
26         Document doc = dBuilder.parse(xmlFile);
27         doc.getDocumentElement().normalize();
28         System.out.println("XML Módosítása");
29         System.out.println("Adja meg mit szeretne módosítani: ");
30         System.out.println("1 Beszállító módosítása\n2 Pizza módosítása\n3 Futár módosítása");
31         Modify(doc);
32     }
33
34

```

```

34
35 public static void ModifyXML(Document doc) throws TransformerException {
36     TransformerFactory transformerFactory = TransformerFactory.newInstance();
37     Transformer transformer = transformerFactory.newTransformer();
38     DOMSource source = new DOMSource(doc);
39     StreamResult result = new StreamResult(new File("XMLmoizhl.xml"));
40     transformer.transform(source, result);
41 }
42
43
44 public static void Modify(Document doc) throws TransformerException {
45     int BeszallitokSzama = doc.getElementsByTagName("Beszallito").getLength(); // beszallitok számának lekérdezése
46     int PizzakSzama = doc.getElementsByTagName("Pizza").getLength(); // pizzak számának lekérdezése
47     int FutarokSzama = doc.getElementsByTagName("Futar").getLength(); // futarok számának lekérdezése
48
49     Scanner scan = new Scanner(System.in);
50     System.out.println("Adja meg a sorszámot: ");
51     int readCategory = scan.nextInt();
52     switch (readCategory) {
53     case 1:
54         ModifyBeszallitok(doc, BeszallitokSzama);
55         break;
56     case 2:
57         ModifyPizzak(doc, PizzakSzama);
58         break;
59     case 3:
60         ModifyFutarok(doc, FutarokSzama);
61         break;
62     }
63     scan.close();
64 }
65
66 public static String ReadId() {
67     Scanner sc = new Scanner(System.in);
68     System.out.print("\nid:");
69     String id = sc.nextLine();
70     return id;
71 }
72
73 private static void ModifyBeszallitok(Document doc, int beszallitoszam) throws TransformerException {
74     // Kiíratjuk a jelenlegi Beszállítókat, majd lekérdezzük melyiket kívánjuk módosítani.
75
76     System.out.println("Melyik Beszállítónak az adatait szeretné módosítani?");
77
78     for (int i = 1; i < beszallitoszam + 1; i++) {
79         System.out.println(i + ". beszállító");
80         DomReadMOIZHL.ReadBeszallitoById(doc, String.valueOf(i));
81         System.out.println("-----");
82     }
83     String id = ReadId();
84     // Bekérjük az új adatokat
85     Scanner sc = new Scanner(System.in);
86     System.out.print("Név: ");
87     String Nev = sc.nextLine();
88     System.out.print("Iranyitoszam: ");
89     String Iranyitoszam = sc.nextLine();
90     System.out.print("Varos: ");
91     String Varos = sc.nextLine();
92     System.out.print("Utca: ");
93     String Utca = sc.nextLine();
94     System.out.print("Hazszam: ");
95     String Hazszam = sc.nextLine();
96     sc.close();

```



```

97 // lekérdezzük az Elementeket, majd setTextContent-el módosítjuk.
98 NodeList nodeList = doc.getElementsByTagName("Beszallito");
99 for (int i = 0; i < nodeList.getLength(); i++) {
100     Node nNode = nodeList.item(i);
101     if (nNode.getNodeType() == Node.ELEMENT_NODE) {
102         Element element = (Element) nNode;
103         String sid = element.getAttribute("BeszallitoID");
104         if (sid.equals(id)) {
105             Node node1 = element.getElementsByTagName("Nev").item(0);
106             node1.setTextContent(Nev);
107             Node node2 = element.getElementsByTagName("Iranyitoszam").item(0);
108             node2.setTextContent(Iranyitoszam);
109             Node node3 = element.getElementsByTagName("Varos").item(0);
110             node3.setTextContent(Varos);
111             Node node4 = element.getElementsByTagName("Utca").item(0);
112             node4.setTextContent(Utca);
113             Node node5 = element.getElementsByTagName("Hatszam").item(0);
114             node5.setTextContent(Hatszam);
115             System.out.println("Sikeres módosítás");
116         }
117     }
118 }
119 ModifyXML(doc); // Létrehozzuk az XML-t
120 }

121 private static void ModifyPizzak(Document doc, int pizzaszam) throws TransformerException {
122     System.out.println("Melyik pizzát kívánja módosítani?");
123     for (int i = 1; i < pizzaszam + 1; i++) {
124         System.out.println(i + ". pizza");
125         DomReadMOIIZHL.ReadPizzaById(doc, String.valueOf(i));
126         System.out.println("-----");
127     }
128     String id = ReadId();
129     // Bekérjük az új adatokat
130     Scanner sc = new Scanner(System.in);
131     System.out.print("Meret: ");
132     String Meret = sc.nextLine();
133     System.out.print("Teljes_ar: ");
134     String Teljes_ar = sc.nextLine();
135     System.out.print("Pizza_neve: ");
136     String Pizza_neve = sc.nextLine();
137     sc.close();
138     // lekérdezzük az Elementeket, majd setTextContent-el módosítjuk.
139     NodeList nodeList = doc.getElementsByTagName("Pizza");
140     for (int i = 0; i < nodeList.getLength(); i++) {
141         Node nNode = nodeList.item(i);
142         if (nNode.getNodeType() == Node.ELEMENT_NODE) {
143             Element element = (Element) nNode;
144             String sid = element.getAttribute("PizzaID");
145             if (sid.equals(id)) {
146                 Node node1 = element.getElementsByTagName("Meret").item(0);
147                 node1.setTextContent(Meret);
148                 Node node2 = element.getElementsByTagName("Teljes_ar").item(0);
149                 node2.setTextContent(Teljes_ar);
150                 Node node3 = element.getElementsByTagName("Pizza_neve").item(0);
151                 node3.setTextContent(Pizza_neve);
152                 System.out.println("Sikeres módosítás");
153             }
154         }
155     }
156     ModifyXML(doc); // Létrehozzuk az XML-t
157 }

```

```

158 private static void ModifyFutarok(Document doc, int futarszam) throws TransformerException {
159     System.out.println("Melyik Futárt kívánja módosítani?");
160     for (int i = 1; i < futarszam + 1; i++) {
161         System.out.println(i + ". futar");
162         DomReadMOIZHL.ReadFutarById(doc, String.valueOf(i));
163         System.out.println("-----");
164     }
165     String id = ReadId();
166     // Bekérjük az új adatokat
167     Scanner sc = new Scanner(System.in);
168     System.out.print("Nev: ");
169     String Nev = sc.nextLine();
170     System.out.print("Telefonszam: ");
171     String Telefonszam = sc.nextLine();
172     sc.close();
173     // lekérdezzük az Elementeket, majd setTextContent-el módosítjuk.
174     NodeList nodeList = doc.getElementsByTagName("Futar");
175     for (int i = 0; i < nodeList.getLength(); i++) {
176         Node nNode = nodeList.item(i);
177         if (nNode.getNodeType() == Node.ELEMENT_NODE) {
178             Element element = (Element) nNode;
179             String sid = element.getAttribute("FutarID");
180             if (sid.equals(id)) {
181                 Node node1 = element.getElementsByTagName("Nev").item(0);
182                 node1.setTextContent(Nev);
183                 Node node2 = element.getElementsByTagName("Telefonszam").item(0);
184                 node2.setTextContent(Telefonszam);
185                 System.out.println("Sikeres módosítás");
186             }
187         }
188     }
189     ModifyXML(doc); // létrehozzuk az XML-t
190 }
191
192
193 }

```