

Jegyzőkönyv

Adatkezelés XML környezetben

Étterem működése

Féléves feladat

Készítette: **Mészáros Marcell**

Nepunkód: **X0R8S7**

A feladat leírása:

Az én feladatomban egy olyan ER modellt kellett készítenem, amellyel egy étterem vezetni tudja a készleten lévő alapanyagait, ételeit, itálait, kiszolgálóit, avagy futárjait az ügyfelekkel és az általuk leadott megrendelésekkel együtt. Az ER modellemben 6 egyed található

Étterem, Alapanyagok, Étél, Vendég, Kiszolgálók, illetve Rendelések

Az Étterem egyed tartalmazza azokat az adatokat, melyeket az adott étteremről tudni kell, úgy mint a Cégjegyzékszám (ez az egyedi azonosító), név, cím és telefonszám.

Az alapanyagok egyed megmutatja, hogy milyen alapanyagok vannak az ételekhez készleten, milyen mennyiségben vannak raktáron és mikor fog elkövetkezni a legelső szavatossági idő, amire a dolgozóknak oda kell figyelniük.

Az étél egyedben a menün lévő készételek találhatóak meg, ezekhez lesznek rendelve a raktáron lévő alapanyagok, és ezek eladásával lehet csökkenteni a készletet. Ami még nagyon fontos, hogy itt van tárolva minden eladható termék (étél) ára is. Itt a készlet egyed származtatott, mivel a készételek hozzávalóinak a mennyiségéből lehet megadni, hogy mennyi étél készülhet el az adott raktári mennyiségből.

A rendelések egyedben vannak a vendégek által az étterem felé leadott megrendelések készételekre, melyeket majd a kiszolgálók segítségével kaphatnak meg. Itt meg kell majd adni a rendelések fizetési módját, illetve a fogyasztást is, hogy helyben fogyasztják el, elviszi a vendég, vagy házhoz kell neki szállítani.

A kiszolgálók egyed mutatja meg, hogy milyen ki-, illetve felszolgálók, valamint futárok dolgoznak, típus szerint megadva, akik a felvett rendeléseket átadják a vendégeknek.

A Vendég egyed tartalmazza minden fontos adatot az étterem vendégeiről, amik szükségesek adott esetben a kiszolgálásukhoz, itt a telefonszám, illetve a cím tulajdonság többértékű, mivel rendelkezhetnek több telefonszámmal is, illetve házhozszállítás esetében nem feltétlen mindig ugyan arra a címre rendelnek, otthon mellett akár munkahelyre is kérhetnek kiszállítást.

Röviden az egyedek közötti kapcsolatokról. Az Étterem és Vendég kapcsolat 1:N, mivel az adott étterem szemszögéből több vendégük is lehet, viszont a szempontjukból irreleváns, hogy másik étterembe is elmehetnek vagy rendelhetnek.

Az Étterem és Étél között szintén 1:N kapcsolat van, az étteremnek több étele is van, viszont a szempontjukból fontos módon azt az ételt csak abban az egy étteremben készítik el.

Étél és Alapanyagok egyedek esetében N:M kapcsolat áll fenn, mivel egy alapanyagok több készételhez is szükséges lehet, illetve egy ételhez is több alapanyagra lehet szükség.

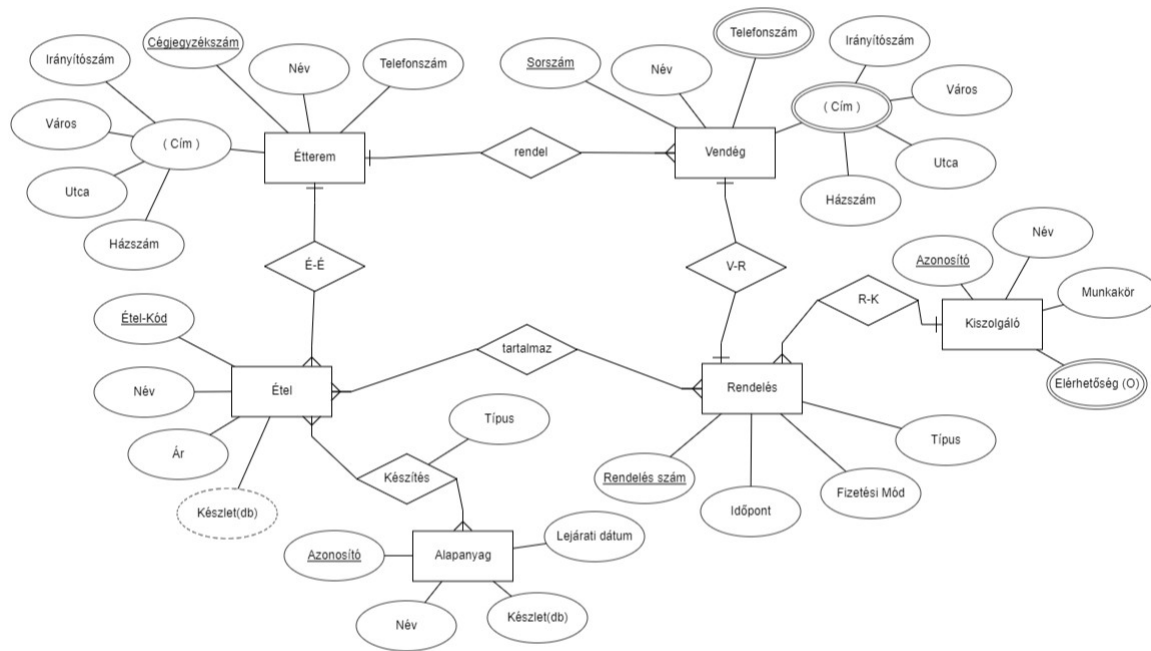
Ezen logika mentén az Étél és Rendelés egyedek között is N:M kapcsolat van, egy rendelésben több étel is szerepelhet, mint ahogy egy étel több rendelésben.

Rendelés, illetve Vendég egyedek esetében már más a helyzet, itt 1:1 kapcsolat van a feladatban először, mivel egy fogyasztás vagy rendelést bizonyosan egy vendéghez tartozik, mint ahogy egy alkalommal egy vendég egyszer is ad le megrendelést egy étterem számára.

Elérkeztünk utolsó kapcsolatunkhoz, ez a Rendelés és a Kiszolgáló egyedek között kötött meg, itt is 1:N kapcsolatról van szó, mivel egy kiszolgáló egyszerre több rendeléssel is el tud bánni, de egy adott megrendelést bizonyosan egyetlen kiszolgáló fog elvégezni.

1. feladat

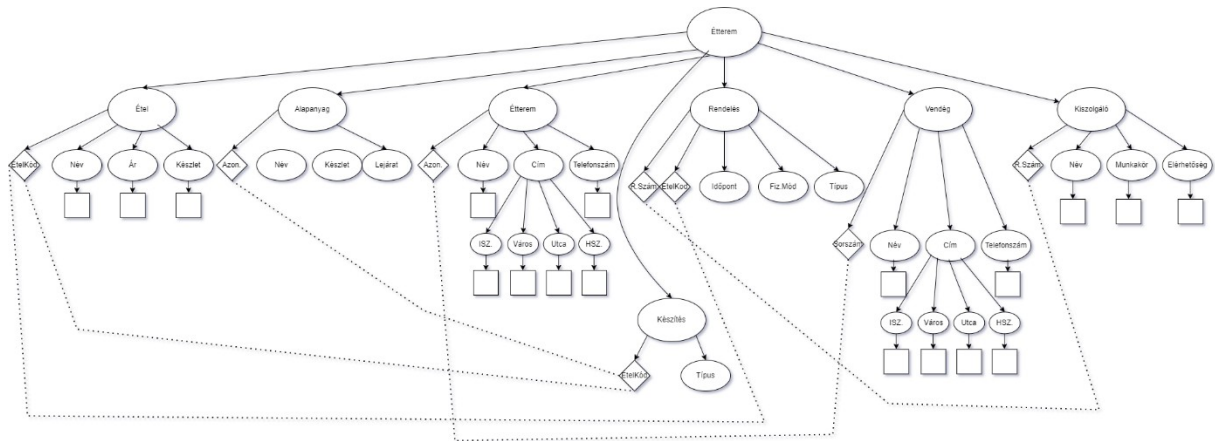
1a, Az adatbázis ER modell:



Az ER modell leírása fentebb látható a feladat részletezésében.

1b, Az adatbázis konvertálása XDM modellre:

Ebben a feladatrészletben elkészítettem a már megalkotott ER modell XDM modellre való konvertálását, ami alább látható is. Ezen modell segítségével könnyedén készíthetünk a feladatra megfelelő XML dokumentumot.



1c, Az XDM modell alapján XML dokumentum készítése:

```
1  <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2  <etterem>
3      <etterem1 kod="1">
4          <nev>Tüzes Pizzéria</nev>
5          <cim>
6              <ir_szam>9876</ir_szam>
7              <varos>Pornóapáti</varos>
8              <utca>Forró utca</utca>
9              <hazszam>11</hazszam>
10         </cim>
11         <telefonszam>06301234567</telefonszam>
12     </etterem1>
13     <alapanyag azon="1">
14         <nev>Sajt</nev>
15         <keszlet>8</keszlet>
16         <lejarat>2022.04.11.</lejarat>
17     </alapanyag>
18     <alapanyag azon="2">
19         <nev>Sonka</nev>
20         <keszlet>6</keszlet>
21         <lejarat>2022.05.25.</lejarat>
22     </alapanyag>
23     <alapanyag azon="3">
24         <nev>Szalámi</nev>
25         <keszlet>2</keszlet>
26         <lejarat>2022.02.25.</lejarat>
27     </alapanyag>
28     <alapanyag azon="4">
29         <nev>Hagyma</nev>
30         <keszlet>10</keszlet>
31         <lejarat>2021.12.25.</lejarat>
32     </alapanyag>
```

```
33     <etel ekod="1" azon="1">
34         <nev>Margaréta</nev>
35         <ar>1890</ar>
36         <keszlet>8</keszlet>
37     </etel>
38     <etel ekod="2" azon="2">
39         <nev>SonkaSajt</nev>
40         <ar>1990</ar>
41         <keszlet>6</keszlet>
42     </etel>
43     <etel ekod="3" azon="3">
44         <nev>SzalamiSajt</nev>
45         <ar>2090</ar>
46         <keszlet>2</keszlet>
47     </etel>
48     <etel ekod="4" azon="4">
49         <nev>HagymaSajt</nev>
50         <ar>1990</ar>
51         <keszlet>8</keszlet>
52     </etel>
53     <kiszolgallo kk="1" rsz="1">
54         <nev>Balogh Mariska</nev>
55         <munkakor>pincer</munkakor>
56         <elerhetoseg>06202345678</elerhetoseg>
57     </kiszolgallo>
58     <kiszolgallo kk="2" rsz="2">
59         <nev>Kiss Gábor</nev>
60         <munkakor>futar</munkakor>
61         <elerhetoseg>06703456789</elerhetoseg>
62     </kiszolgallo>
```

```
63         <rendeles rsz="1">
64         <idopont>2021.11.11 11:11</idopont>
65         <fizmod>kp</fizmod>
66         <tipus>elvitel</tipus>
67     </rendeles>
68     <rendeles rsz="2">
69     <idopont>2021.11.11 12:11</idopont>
70     <fizmod>bk</fizmod>
71     <tipus>kiszallitas</tipus>
72 </rendeles>
73     <rendeles rsz="3">
74     <idopont>2021.11.11 13:11</idopont>
75     <fizmod>szepkartya</fizmod>
76     <tipus>helybenfogy</tipus>
77 </rendeles>
78 <vendeg sorszam="1">
79     <nev>Nagy Bernadett</nev>
80     <cim>
81         <ir_szam>9876</ir_szam>
82         <varos>Pornóapáti</varos>
83         <utca>Hideg utca</utca>
84         <hazszam>1</hazszam>
85     </cim>
86     <telefonszam>06309876654</telefonszam>
87 </vendeg>
```

```
88     <vendeg sorszam="2">
89         <nev>Nagy János</nev>
90         <cim>
91             <ir_szam>9875</ir_szam>
92             <varos>Háromhányástanya</varos>
93             <utca>Alkohol</utca>
94             <hazszam>99</hazszam>
95         </cim>
96         <telefonszam>06505551289</telefonszam>
97     </vendeg>
98     <vendeg sorszam="3">
99         <nev>Pici Béla</nev>
100        <cim>
101            <ir_szam>9877</ir_szam>
102            <varos>Dunavarsány</varos>
103            <utca>Ivó utca</utca>
104            <hazszam>33</hazszam>
105        </cim>
106        <telefonszam>06705896455</telefonszam>
107    </vendeg>
108    <keszites>
109        <tipus>sutes</tipus>
110    </keszites>
111 </etterem>
```

1d, Az XML dokumentum alapján XMLSchema készítése:


```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
3
4      <xs:element name="etterem">
5          <xs:complexType>
6              <xs:sequence>
7                  <xs:choice maxOccurs="unbounded">
8                      <xs:element name="etterem1" type="etterem_tipus" />
9                      <xs:element name="alapanyag" type="alapanyag_tipus" />
10                     <xs:element name="etel" type="etel_tipus" />
11                     <xs:element name="kiszolgallo" type="kiszolgallo_tipus" />
12                     <xs:element name="rendeles" type="rendeles_tipus" />
13                     <xs:element name="vendeg" type="vendeg_tipus" />
14                     <xs:element name="keszites" type="keszites_tipus" />
15                 </xs:choice>
16             </xs:sequence>
17         </xs:complexType>
18         <xs:key name="kod">
19             <xs:selector xpath="etterem1" />
20             <xs:field xpath="@kod" />
21         </xs:key>
22         <xs:key name="azon">
23             <xs:selector xpath="alapanyag" />
24             <xs:field xpath="@azon" />
25         </xs:key>
26         <xs:key name="ekod">
27             <xs:selector xpath="etel" />
28             <xs:field xpath="@ekod" />
29         </xs:key>
30         <xs:key name="kk">
31             <xs:selector xpath="kiszolgallo" />
32             <xs:field xpath="@kk" />
```

```
33     </xs:key>
34     <xs:key name="rsz">
35         <xs:selector xpath="rendeles" />
36         <xs:field xpath="@rsz" />
37     </xs:key>
38 </xs:element>
39
40 <xs:complexType name="etterem1_tipus">
41     <xs:sequence>
42         <xs:element name="nev" type="nev_tipus" />
43         <xs:element name="cim" type="cim_tipus" />
44         <xs:element name="telefonszam" type="telefonszam_tipus" />
45     </xs:sequence>
46     <xs:attribute type="xs:string" name="kod" use="required" />
47 </xs:complexType>
48
49 <xs:complexType name="alapanyag_tipus">
50     <xs:sequence>
51         <xs:element name="nev" type="nev_tipus" />
52         <xs:element name="keszlet" type="keszlet_tipus" />
53         <xs:element name="lejarat" type="lejarat_tipus" />
54     </xs:sequence>
55     <xs:attribute type="xs:string" name="azon" use="required" />
56     <xs:attribute type="xs:string" name="ekod" use="required" />
57 </xs:complexType>
58
59 <xs:complexType name="etel_tipus">
60     <xs:sequence>
61         <xs:element name="nev" type="elerhetoseg_tipus" />
62         <xs:element name="ar" type="ar_tipus" />
63         <xs:element name="keszlet" type="keszlet_tipus" />
64     </xs:sequence>
65     <xs:attribute type="xs:string" name="ekod" use="required" />
66 </xs:complexType>
```

```
67
68     <xs:complexType name="kiszolgalo_tipus">
69         <xs:sequence>
70             <xs:element name="nev" type="nev_tipus" />
71             <xs:element name="munkakor" type="munkakor_tipus" />
72             <xs:element name="elerhetoseg" type="elerhetoseg_tipus" />
73         </xs:sequence>
74         <xs:attribute type="xs:string" name="kk" use="required" />
75         <xs:attribute type="xs:string" name="rsz" use="required" />
76     </xs:complexType>
77
78     <xs:complexType name="rendeles_tipus">
79         <xs:sequence>
80             <xs:element name="idopont" type="idopont_tipus" />
81             <xs:element name="fizmod" type="fizmod_tipus" />
82             <xs:element name="tipus" type="tipus_tipus" />
83         </xs:sequence>
84         <xs:attribute type="xs:string" name="rsz" use="required" />
85     </xs:complexType>
86
87     <xs:complexType name="vendeg_tipus">
88         <xs:sequence>
89             <xs:element name="nev" type="nev_tipus" />
90             <xs:element name="cim" type="cim_tipus" />
91             <xs:element name="telefonszam" type="telefonszam_tipus" />
92         </xs:sequence>
93     </xs:complexType>
94
95     <xs:complexType name="vendeg_tipus">
96         <xs:sequence>
97             <xs:element name="vendeg_szam" type="vendeg_szam_tipus" />
98         </xs:sequence>
99     </xs:complexType>
```

```
100
101     <xs:complexType name="cim_tipus">
102         <xs:sequence>
103             <xs:element name="ir_szam" type="ir_szam_tipus" />
104             <xs:element name="varos" type="varos_tipus" />
105             <xs:element name="utca" type="utca_tipus" />
106             <xs:element name="hazszam" type="hazszam_tipus" />
107         </xs:sequence>
108     </xs:complexType>
109
110     <xs:simpleType name="nev_tipus">
111         <xs:restriction base="xs:string">
112             <xs:minLength value="4" />
113         </xs:restriction>
114     </xs:simpleType>
115
116     <xs:simpleType name="fizmod_tipus">
117         <xs:restriction base="xs:string">
118             <xs:minLength value="2" />
119         </xs:restriction>
120     </xs:simpleType>
121
122     <xs:simpleType name="etel_tipus">
123         <xs:restriction base="xs:string">
124             <xs:minLength value="2" />
125         </xs:restriction>
126     </xs:simpleType>
127
128     <xs:simpleType name="ital_tipus">
129         <xs:restriction base="xs:string">
130             <xs:minLength value="2" />
131         </xs:restriction>
132     </xs:simpleType>
133
```

```
133
134     <xs:simpleType name="elerhetoseg_tipus">
135         <xs:restriction base="xs:integer">
136             <xs:minInclusive value="1" />
137         </xs:restriction>
138     </xs:simpleType>
139
140     <xs:simpleType name="munkakor_tipus">
141         <xs:restriction base="xs:string">
142             <xs:minLength value="3" />
143         </xs:restriction>
144     </xs:simpleType>
145
146     <xs:simpleType name="tipus_tipus">
147         <xs:restriction base="xs:string">
148             <xs:minLength value="3" />
149         </xs:restriction>
150     </xs:simpleType>
151
152     <xs:simpleType name="utca_tipus">
153         <xs:restriction base="xs:string">
154             <xs:minLength value="3" />
155         </xs:restriction>
156     </xs:simpleType>
157
158
159     <xs:simpleType name="keszlet_tipus">
160         <xs:restriction base="xs:integer">
161             <xs:minInclusive value="1" />
162         </xs:restriction>
163     </xs:simpleType>
164
```

```
164
165     <xs:simpleType name="hazsam_tipus">
166         <xs:restriction base="xs:integer">
167             <xs:minInclusive value="1" />
168         </xs:restriction>
169     </xs:simpleType>
170
171     <xs:simpleType name="ir_szam_tipus">
172         <xs:restriction base="xs:integer">
173             <xs:minInclusive value="1" />
174         </xs:restriction>
175     </xs:simpleType>
176
177     <xs:simpleType name="ar_tipus">
178         <xs:restriction base="xs:integer">
179             <xs:minInclusive value="2" />
180         </xs:restriction>
181     </xs:simpleType>
182
183     <xs:simpleType name="lejarat_tipus">
184         <xs:restriction base="xs:integer">
185             <xs:minInclusive value="1" />
186         </xs:restriction>
187     </xs:simpleType>
188
189     <xs:simpleType name="idopont_tipus">
190         <xs:restriction base="xs:integer">
191             <xs:minInclusive value="1" />
192         </xs:restriction>
193     </xs:simpleType>
194 </xs:schema>
```

2. feladat

2a, Adatolvasás:

Ebben a feladatrészletben egy olyan java programkódot készítettem, amely a meglévő XML dokumentumunkból tudja a benne tárolt adatokat kiolvasni.

```
package hu.domparse.x0r8s7;

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

public class DOMReadx0r8s7 {

    public static void main(String[] args) {
        try {
            File xmlFile = new File("XMLx0r8s7.xml"); // fájl beolvasása
            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
            DocumentBuilder dBuilder = factory.newDocumentBuilder();
            Document doc = dBuilder.parse(xmlFile);
            doc.getDocumentElement().normalize();
            Read(doc);
            //kivételkezelés
        } catch (ParserConfigurationException pce) {
            pce.printStackTrace();
        } catch (IOException ioe) {
            ioe.printStackTrace();
        } catch (SAXException sae) {
            sae.printStackTrace();
        }
    }

    public static void Read(Document doc) {
        NodeList nList = doc.getElementsByTagName("etterem1"); //NoteList létrehozása,
        amiben tároljuk az elemeket
        for (int i = 0; i < nList.getLength(); i++) {
```

```

        Node nNode = nList.item(i);
        Element element = (Element) nNode;
        if (nNode.getNodeType() == Node.ELEMENT_NODE) { //az azonosító kódok
            String kod = element.getAttribute("kod");
            String azon = element.getAttribute("azon");
            String kk = element.getAttribute("kk");
            ReadEtteremById(doc, kod);
            ReadAlapanyagById(doc, azon);
            ReadKiszolgalloById(doc, kk);
        }
    }
}

public static void ReadEtteremById(Document doc, String kod) {
    NodeList nList = doc.getElementsByTagName("etterem1"); //étterem adatait
    tartalmazó nodeList létrehozása
    for (int i = 0; i < nList.getLength(); i++) {
        Node nNode = nList.item(i);
        Element element = (Element) nNode;
        if (nNode.getNodeType() == Node.ELEMENT_NODE) {
            kiválasztása
            if (element.getAttribute("kod").equals(kod)) { //kiírandó adatok
                String telefonszam =
                element.getElementsByTagName("telefonszam").item(0).getTextContent();
                String etel =
                element.getElementsByTagName("etel").item(0).getTextContent();
                String nev =
                element.getElementsByTagName("nev").item(0).getTextContent();
                String cim =
                element.getElementsByTagName("cim").item(0).getTextContent();
                System.out.println(" Az étterem adatai: \n\tNév:\t" + nev
                    + "\n\tTelefonszám:\t" + telefonszam + "\n\tÉtelek:\t" + etel + "\n\tCím:\t" + cim); //kiíratás
            }
        }
    }
}

public static void ReadAlapanyagById(Document doc, String azon) {
    NodeList nList = doc.getElementsByTagName("alapanyag"); //az alapanyagokat
    tartalmazó nodeList létrehozása
    for (int i = 0; i < nList.getLength(); i++) {
        Node nNode = nList.item(i);
        Element element = (Element) nNode;
        if (nNode.getNodeType() == Node.ELEMENT_NODE) {

```



```

        if (element.getAttribute("azon").equals(bid)) { //kiírandó
adatok kiválasztása
            String nev =
element.getElementsByTagName("nev").item(0).getTextContent();
            String keszlet =
element.getElementsByTagName("keszlet").item(0).getTextContent();
            String lejarat =
element.getElementsByTagName("lejarat").item(0).getTextContent();
            System.out.println("Alapanyagok: \n\tNév:\t" + nev + "\n\t"
tkeszlet:\t" + keszlet
                                + "\n\tLejárat:\t" + lejarat); //kiíratás
            String ekod = element.getAttribute("ekod");
            ReadTermeloById(doc, ekod);
        }
    }
}

```

```

public static void ReadEtelById(Document doc, String ekod) {
    NodeList nList = doc.getElementsByTagName("etel"); //a készételek adatait
    tartalmazó nodeList létrehozása
    for (int i = 0; i < nList.getLength(); i++) {
        Node nNode = nList.item(i);
        Element element = (Element) nNode;
        if (nNode.getNodeType() == Node.ELEMENT_NODE) {
adatok kiválasztása
            if (element.getAttribute("ekod").equals(ekod)) { //kiírandó
                String nev =
element.getElementsByTagName("nev").item(0).getTextContent();
                String ar =
element.getElementsByTagName("ar").item(0).getTextContent();
                String keszlet =
element.getElementsByTagName("keszlet").item(0).getTextContent();
                System.out.println("Készétel: \n\tNév:\t" + nev + "\n\t"
tÁr:\t" + ar
                                + "\n\tKészlet:\t" + keszlet); //kiíratás
            }
        }
    }
}

```

```

public static void ReadVendegById(Document doc, String sorszam) {
    NodeList nList = doc.getElementsByTagName("vendeg"); //a vendégek adatait
    tartalmazó nodeList létrehozása
    for (int i = 0; i < nList.getLength(); i++) {
        Node nNode = nList.item(i);
        Element element = (Element) nNode;

```

```

        if (nNode.getNodeType() == Node.ELEMENT_NODE) {
            if (element.getAttribute("sorszam").equals(sorszam)) {
//kiírandó adatok kiválasztása
                String nev =
element.getElementsByTagName("nev").item(0).getTextContent();
                String telefonszam =
element.getElementsByTagName("telefonszam").item(0).getTextContent();
                System.out.println("Személyzet adatok: \n\tNév:\t" + nev
+ "\n\tTelefonszám:\t" + telefonszam
                                + "\n\tCím:\t" + cim + "\n\tMunkakör:\t" +
munkakor) //kiíratás
                String rsz = element.getAttribute("rsz");
                ReadFoglalasById(doc, rsz);
            }
        }
    }
}

```

```

public static void ReadRendelesById(Document doc, String rsz) {
    NodeList nList = doc.getElementsByTagName("rendeles"); //a rendelések adatait
    tartalmazó nodeList létrehozása
    for (int i = 0; i < nList.getLength(); i++) {
        Node nNode = nList.item(i);
        Element element = (Element) nNode;
        if (nNode.getNodeType() == Node.ELEMENT_NODE) {
            if (element.getAttribute("rsz").equals(fid)) { //kiírandó adatok
//kiválasztása
                String idopont =
element.getElementsByTagName("idopont").item(0).getTextContent();
                String fizmod =
element.getElementsByTagName("fizmod").item(0).getTextContent();
                String tipus =
element.getElementsByTagName("tipus").item(0).getTextContent();
                System.out.println("Rendelés adatok: \n\tIdőpont:\t" +
idopont + "\n\tFizetési Mód:\t" + fizmod + "\n\tTípus:\t" + tipus); //kiíratás
            }
        }
    }
}
}

```

```

// #CreatedByMeszarosMarcell

```

2b, Adatmódosítás:

Ebben a feladatrészletben egy olyan java programkódot készítettem, amely a meglévő XML dokumentumunkban tudja a benne tárolt adatokat más általunk megadott adatra változtatni.

```
package hu.domparse.x0r8s7;

import org.w3c.dom.Element;
import org.w3c.dom.Document;
import org.w3c.dom.NamedNodeMap;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import java.io.File;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;

public class DOMModifyx0r8s7 {

    public static void main(String[] args) {
        try {
            File inputFile = new File("XMLx0r8s7.xml");
            DocumentBuilderFactory docFactory =
DocumentBuilderFactory.newInstance();

            DocumentBuilder docBuilder = docFactory.newDocumentBuilder();
            Document doc = docBuilder.parse(inputFile);
            Node vendeg = doc.getElementsByTagName("vendeg").item(0);

            NamedNodeMap attr = vendeg.getAttributes();                // 10-re
módosítjuk a sorszam-t 1-ről
            Node nodeAttr = attr.getNamedItem("sorszam");
            nodeAttr.setTextContent("10");

            NodeList list = vendeg.getChildNodes();

            for (int temp = 0; temp < list.getLength(); temp++) {
                // a hibaüzenet megváltoztatása
                Node node = list.item(temp);
```

```

        if (node.getNodeType() == Node.ELEMENT_NODE) {
            Element hElement = (Element) node;
            if ("sorszam".equals(hElement.getNodeName())) {
                if ("123456BC".equals(hElement.getTextContent()))
                    hElement.setTextContent("123456BC");
            }
        }
    }

    NodeList list2 = doc.getElementsByTagName("alapanyag");
    for (int j = 0; j < list2.getLength(); j++) {
        Node alapanyag = list2.item(j);
        if (alapanyag.getNodeType() == Node.ELEMENT_NODE) {
            String id =
alapanyag.getAttributes().getNamedItem("azon").getTextContent();
            if ("1".equals(id.trim())) {
                NodeList gyerekNodes = alapanyag.getChildNodes();
                for (int k = 0; k < gyerekNodes.getLength(); k++)
                {
                    Node item = gyerekNodes.item(k);
                    if (item.getNodeType() ==
Node.ELEMENT_NODE) { // kitöröljük az vendégből a kód mezőt
                        if
("telefonszam".equalsIgnoreCase(item.getNodeName())) {
                            alapanyag.removeChild(item);
                        }
                        // módosítjuk az 1-es sorszam-mal
                        if
("nev".equalsIgnoreCase(item.getNodeName())) {
                            item.setTextContent("ABC");
                        }
                    }
                }
            }
        }
    }

    // kiíratás
    TransformerFactory transformerFactory =
TransformerFactory.newInstance();
    Transformer transformer = transformerFactory.newTransformer();
    DOMSource source = new DOMSource(doc);
    System.out.println("-----Módosított File-----");
    StreamResult consoleResult = new StreamResult(System.out);
    transformer.transform(source, consoleResult);

```

```

        } catch (Exception e) {
            e.printStackTrace();
        }

    }
}

```

2c, Adat lekérdezés:

Ebben a feladatrészletben egy olyan java programkódot készítettem, amely a meglévő XML dokumentumunkból tudja a benne tárolt adatokat lekérdezni.

```

package hu.domparse.x0r8s7;

import java.io.File;
import java.util.Scanner;
import java.io.IOException;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.transform.TransformerException;

import org.w3c.dom.Element;
import org.w3c.dom.Document;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;
import javax.xml.parsers.ParserConfigurationException;

public class DOMQueryx0r8s7 {

    public static void main(String[] args)
        throws ParserConfigurationException, IOException, SAXException,
        TransformerException {
        File xmlFile = new File("XMLx0r8s7.xml"); //beolvassuk a fájlt
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = factory.newDocumentBuilder();
        Document doc = dBuilder.parse(xmlFile);
        doc.getDocumentElement().normalize();
        LoadtermeloQuery(doc);
    }

    public static void LoadtermeloQuery(Document doc) throws TransformerException {
        NodeList nodeList = doc.getElementsByTagName("etel"); //etel nodeList létrehoz
        String etel;
        Element element = null;
        Node nNode = null;
        for (int i = 0; i < nodeList.getLength(); i++) {
            nNode = nodeList.item(i);
            element = (Element) nNode;
            String nev =
            element.getElementsByTagName("nev").item(0).getTextContent(); //név szerint lekérdezzük
            System.out.println(nev);
        }
        System.out.println("\nÍrja be annak az ételnek a nevét, amelyiknek adatait
szeretné látni:");
        Scanner sc = new Scanner(System.in); //consolról beolvas
        termelo = sc.nextLine();
        for (int i = 0; i < nodeList.getLength(); i++) {
            nNode = nodeList.item(i);
            element = (Element) nNode;

```

```

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

            if (etel.equals("Margaréta")) {
                LoadEtelQuery(doc, "1");
                break;
            }

            if (etel.equals("Sonkasajt")) {
                LoadEtelQuery(doc, "2");
                break;
            }

            if (etel.equals("Szalámisajt")) {
                LoadEtelQuery(doc, "3");
                break;
            }

            if (etel.equals("Hagymasajt")) {
                LoadEtelQuery(doc, "4");
                break;
            }

        }
    }
    sc.close();
}

public static void LoadEtelQuery(Document doc, String ekod) throws
TransformerException {
    NodeList nodeList = doc.getElementsByTagName("etel"); //etel noteList
    létrehozása

    for (int i = 0; i < nodeList.getLength(); i++) {
        Node nNode = nodeList.item(i);
        Element element = (Element) nNode;
        String ekod = element.getAttribute("ekod"); //az ekod alapján keressük
        az elemet

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

            if (ekod.equals(azon)) {
                String Bid = element.getAttribute("ekod"); //ekod alapján
                kivesszük az elemet
                DOMReadx0r8s7.ReadEtelById(doc, ekod); //meghívjuk az
                etel függvényt
            }
        }
    }
}

//CreatedByMeszarosMarcell

```