

Jegyzőkönyv

Adatkezelés XML környezetben

Féléves feladat

Étteremnél leadott rendeléseket kezelő adatbázis

Készítette: Juhász Ákos

Neptunkód: F58KQ8

Dátum: 2022.11.24

Tartalomjegyzék

Feladat leírása	3
1. feladat	3
1a) Az adatbázis ER modell.....	3
1b) Az adatbázis konvertálása XDM modellre	5
1c) Az XDM modell alapján XML dokumentum készítése	5
1d) Az XML dokumentum alapján XMLSchema készítése	9
2. feladat	12
2a) Adatolvasás – DOMRead.....	12
2b) Adatmódosítás – DOMModify	16
2c) Adatlekérdezés – DOMQuery.....	20

Feladat leírása

A feladat egy hétköznapi étteremhez bejövő, telefonos híváson keresztül leadott rendeléseket kezeli.

Magának a rendelésnek letároljuk a leadási időpontját, a vevő által választott kiszállítási címet, és a rendelt ételek fizetendő összértékét.

Természetesen, magukat a megrendelt és megrendelhető ételeket is nyilvántartjuk az adatbázisban. Az ételek tárolva vannak azok nevével, árával, típusával (mint például: pizzák, húsételek, saláták... stb) és a több lehetséges allergénnel, amit az étel tartalmazhat.

A megrendelés és az ételek között a könnyebb áttekinthetőség érdekében, ha egy ételből több is rendelésre kerül, azok mennyiségét feltüntetjük az adatbázisban.

Bizonyos ételek, az év folyamán leárazásra kerülnek, hogy csábítsák a vevőket. Ezeknek az akciós időszakoknak letároljuk a kezdetét, végét és természetesen százalékos értékben a leárazás mértékét is.

A megrendelést leadó vendéget, megrendelőként tartalmazza az adatbázis. Innen lekérdezhetjük a telefonszámát, nevét és, hogy korábban vásárolt-e már az üzletnek valamelyikénél.

A sikeres szállítást követően a fizetés adatai is felkerülnek az adatbázisba. Ezekhez tartozik az átvétel ideje, az hogy a vevő bankkártyával, készpénzzel, vagy esetleg másmilyen fizetőeszközzel rendezte a számlát, és hogy a vásárlónál volt-e esetleg valamilyen általunk kibocsátott kupon fizetéskor.

1. feladat

1a) Az adatbázis ER modell

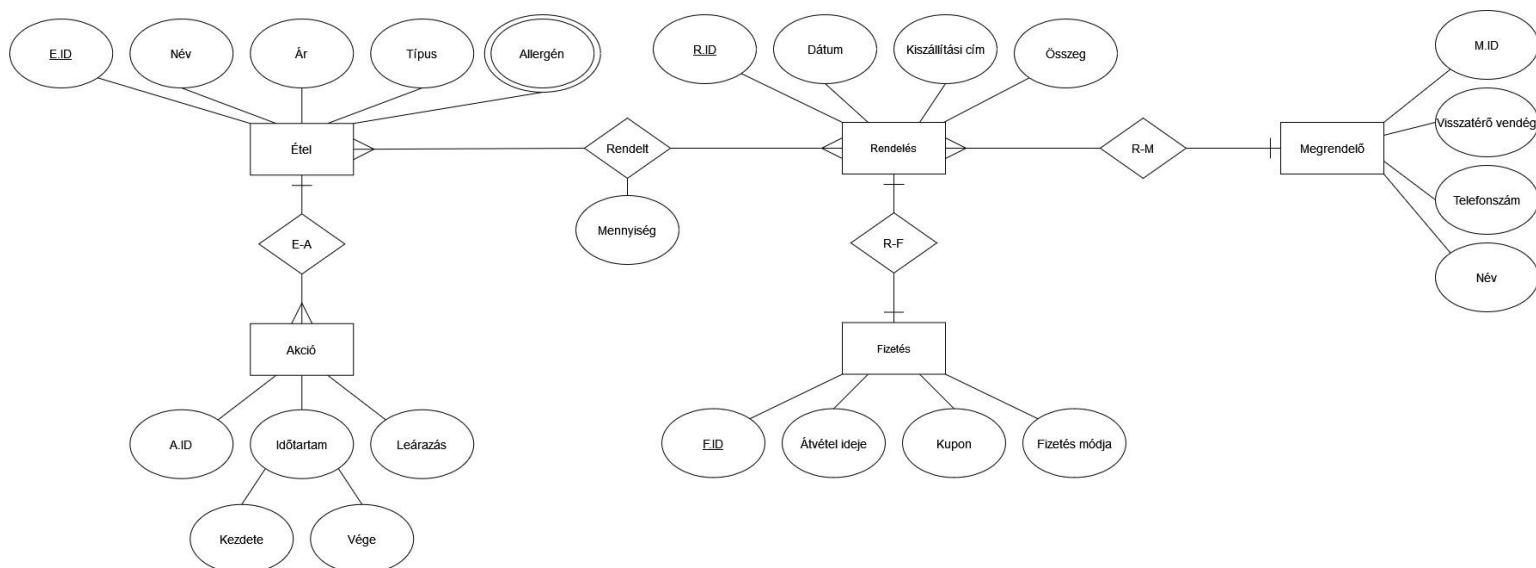
Az adatbázis ER modellre való konvertálása során létrehozunk 5 egyedet (ÉTEL, AKCIÓ, RENDELÉS, FIZETÉS, MEGRENDELŐ).

A rendelés egyed a megrendelővel az R-M kapcsolat köti össze egy egy-több kapcsolatként. Egy rendelést, mivel csak egyszer fizethetünk ki, ezért közöttük egy egy-egy kapcsolat jön létre. Az ételek és rendelések között több-több kapcsolat van, egy további 'mennyiség' tulajdonsággal. Az ételek és akciók között megint egy egy-több kapcsolat van, hiszen egy étel többször is le lehet árazva.

Az ER modellben látható 'allergén' többértékű tulajdonság és az 'időtartam' nevű tulajdonság összetett tulajdonság. Ezeken felül minden egyed el van látva a saját kulcs azonosító tulajdonságával is.

Az egyedek felépítése és tulajdonságaik:

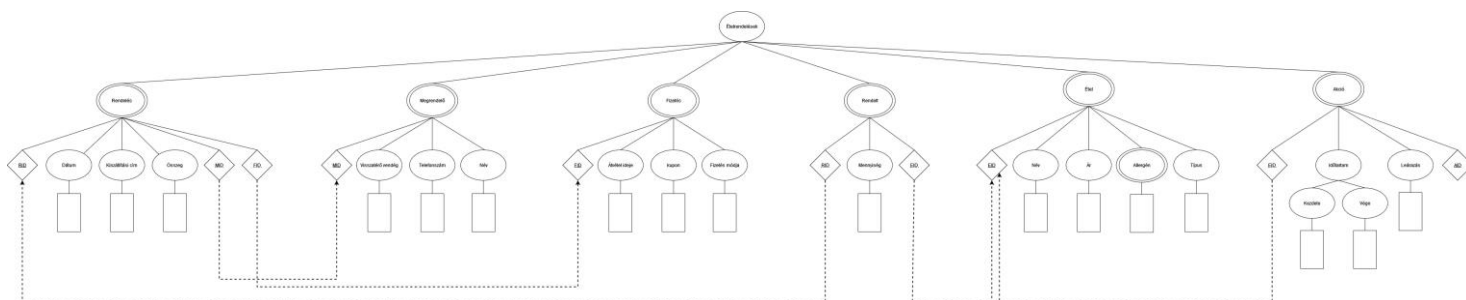
- Rendelés:
 - R.ID: a rendelésekhez tartozó elsődleges kulcs.
 - Dátum: a rendelés leadásának időpontja.
 - Kiszállítási cím: a kiszállítás célcíme.
 - Összeg: a rendelt ételek összesített pénzösszege.
- Étel:
 - E.ID: az ételekhez tartozó elsődleges kulcs.
 - Név: az adott étel neve.
 - Ár: egy adag étel ára.
 - Típus: az étel kategóriája, pl: húsétel, saláta...
 - Allergén: az ételekben előforduló lehetséges allergének.
- Akció:
 - A.ID: az akciókhoz tartozó elsődleges kulcs.
 - Időtartam:
 - Kezdet: az akció kezdeti dátuma.
 - Vége: az akció lejáratának dátuma.
 - Leárazás: százalékos értékben megadva az akció értéke.
- Megrendelő:
 - M.ID: a megrendelőkhez tartozó elsődleges kulcs.
 - Visszatérő vendég: igen/nem érték. Megadja, hogy a vendég vásárolt-e már az egyik üzletben.
 - Telefonszám: a vendég telefonszáma, amiről a rendelése érkezett.
 - Név: a megrendelő neve.
- Fizetés:
 - F.ID: a fizetésekhez tartozó elsődleges kulcs.
 - Átvétel ideje: a kiszállítás után, a rendelés átvételének ideje.
 - Kupon: igen/nem érték. A vásárló használt-e valamilyen, a cég által kibocsátott kupont fizetéskor.
 - Fizetés módja: a fizetés módja pl: készpénz, bankkártya...
- Rendelt kapcsolat:
 - Mennyiség: egy adott ételből rendelt mennyiség.



1b) Az adatbázis konvertálása XDM modellre

Az előzőekben tárgyaltak mellett, az XDM modell kialakítása közben a következő változtatások mentek végbe:

- A 'rendelt' kapcsolatból (Rendelés és Ételek egyed között) elem lett, két másodlagos kulccsal, amik a két táblára mutatnak, és a 'mennyiség' tulajdonsággal.
- Minden elem megkapta a kapcsolatokhoz szükséges másodlagos kulcsait.



1c) Az XDM modell alapján XML dokumentum készítése

Az XDM modell elemeit legalább 3 példányban átkonvertáltam az XML dokumentumba. Az elsődleges és másodlagos kulcsokból, illetve a mennyiség tulajdonságból attribútumok lettek.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <ételrendelések xmlns:xs="http://www.w3.org/2001/XMLSchema-instance" xs:noNamespaceSchemaLocation="XMLSchemaF58KQ8.xsd">
4   <!-- rendelések -->
5   <rendelés RID="1" MID="1" FID="1">
6     <dátum>2021-01-01</dátum>
7     <kiszállítási_cím>Arany_János_utca_1.</kiszállítási_cím>
8     <összeg>6750</összeg>
9   </rendelés>
10
11   <rendelés RID="2" MID="2" FID="2">
12     <dátum>2022-02-02</dátum>
13     <kiszállítási_cím>Nagy_Lasjos_király_útja_2.</kiszállítási_cím>
14     <összeg>6060</összeg>
15   </rendelés>
16
17   <rendelés RID="3" MID="1" FID="3">
18     <dátum>2021-10-10</dátum>
19     <kiszállítási_cím>Arany_János_utca_1.</kiszállítási_cím>
20     <összeg>10880</összeg>
21   </rendelés>
22
23   <rendelés RID="4" MID="3" FID="4">
24     <dátum>2022-11-04</dátum>
25     <kiszállítási_cím>Kuruc_utca_23.</kiszállítási_cím>
26     <összeg>9470</összeg>
27   </rendelés>
28
29   <!-- megrendelők -->
30   <megrendelő MID="1">
31     <visszatérő_vendég>igen</visszatérő_vendég>
32     <telefonszám>0620000001</telefonszám>
33     <név>Első_Elemér</név>
34   </megrendelő>
35
36   <megrendelő MID="2">
37     <visszatérő_vendég>nem</visszatérő_vendég>
38     <telefonszám>0620202020</telefonszám>
39     <név>Második_Mária</név>
40   </megrendelő>
41
42   <megrendelő MID="3">
43     <visszatérő_vendég>nem</visszatérő_vendég>
44     <telefonszám>0620987654321</telefonszám>
45     <név>Kis_Pista</név>
46   </megrendelő>
47
48   <!-- fizetések -->
49   <fizetés FID="1">
50     <átvitel_ideje>2021-01-01</átvitel_ideje>
51     <kupon>nem</kupon>
52     <fizetés_módja>készpénz</fizetés_módja>
53   </fizetés>
54
55   <fizetés FID="2">
56     <átvitel_ideje>2022-02-09</átvitel_ideje>
57     <kupon>igen</kupon>
58     <fizetés_módja>bankkártya</fizetés_módja>
59   </fizetés>
60
61   <fizetés FID="3">
62     <átvitel_ideje>2021-10-11</átvitel_ideje>
63     <kupon>igen</kupon>
64     <fizetés_módja>készpénz</fizetés_módja>
65   </fizetés>
66
67   <fizetés FID="4">
68     <átvitel_ideje>2022-11-07</átvitel_ideje>
69     <kupon>nem</kupon>
70     <fizetés_módja>szép-kártya</fizetés_módja>
71   </fizetés>
72

```

```

72  <!-- megrendelt ételek kapcsoló -->
73  <rendelt RID="1" EID="1">
74    <menyiség>2</menyiség>
75  </rendelt>
76
77  <rendelt RID="1" EID="6">
78    <menyiség>1</menyiség>
79  </rendelt>
80
81  <rendelt RID="2" EID="3">
82    <menyiség>1</menyiség>
83  </rendelt>
84
85  <rendelt RID="2" EID="5">
86    <menyiség>2</menyiség>
87  </rendelt>
88
89  <rendelt RID="3" EID="4">
90    <menyiség>4</menyiség>
91  </rendelt>
92
93  <rendelt RID="4" EID="2">
94    <menyiség>3</menyiség>
95  </rendelt>
96
97  <rendelt RID="4" EID="3">
98    <menyiség>1</menyiség>
99  </rendelt>
100
101  <!-- ételek -->
102  <étel EID="1">
103    <név>Hawaii</név>
104    <ár>2250</ár>
105    <allergén>Glutén</allergén>
106    <allergén>Tojás</allergén>
107    <típus>Pizza</típus>
108  </étel>
109
110  <étel EID="2">
111    <név>Piedone</név>
112    <ár>2250</ár>
113    <allergén>Glutén</allergén>
114    <allergén>Tojás</allergén>
115    <típus>Pizza</típus>
116  </étel>
117
118  <étel EID="3">
119    <név>Szezámcsirke_box</név>
120    <ár>2720</ár>
121    <allergén>Szezámmag</allergén>
122    <típus>Húsetel</típus>
123  </étel>
124
125  <étel EID="4">
126    <név>Gyros_tál</név>
127    <ár>2720</ár>
128    <allergén>Szezámmag</allergén>
129    <allergén>Glutén</allergén>
130    <típus>Húsetel</típus>
131  </étel>
132
133  <étel EID="5">
134    <név>Görög_saláta</név>
135    <ár>1670</ár>
136    <allergén></allergén>
137    <típus>Saláta</típus>
138  </étel>
139
140  <étel EID="6">
141    <név>4sajtos</név>
142    <ár>2550</ár>
143    <allergén>Glutén</allergén>
144    <allergén>Tojás</allergén>
145    <típus>Pizza</típus>
146  </étel>
147

```

```
149 <!-- akciók -->
150 <akció AID="1" EID="1">
151   <időtartam>
152     <kezdet>2022-01-01</kezdet>
153     <vége>2022-02-01</vége>
154   </időtartam>
155   <leírás>10</leírás>
156 </akció>
157
158 <akció AID="2" EID="1">
159   <időtartam>
160     <kezdet>2022-06-01</kezdet>
161     <vége>2022-09-01</vége>
162   </időtartam>
163   <leírás>10</leírás>
164 </akció>
165
166 <akció AID="3" EID="2">
167   <időtartam>
168     <kezdet>2022-10-31</kezdet>
169     <vége>2022-11-30</vége>
170   </időtartam>
171   <leírás>50</leírás>
172 </akció>
173
174 <akció AID="4" EID="6">
175   <időtartam>
176     <kezdet>2022-04-15</kezdet>
177     <vége>2022-05-15</vége>
178   </időtartam>
179   <leírás>15</leírás>
180 </akció>
181
182 <akció AID="5" EID="6">
183   <időtartam>
184     <kezdet>2022-09-01</kezdet>
185     <vége>2022-11-30</vége>
186   </időtartam>
187   <leírás>30</leírás>
188 </akció>
189 </ételrendelések>
```


1d) Az XML dokumentum alapján XMLSchema készítése

Az XML dokumentum validálására létrehoztam az XMLSchema-t. Az XMLSchema elején látható több complex típus, saját egyedként megvalósítva. A ref, key, keyref használatával létrehoztam, az elsődleges és másodlagos kulcsokat az XML fájlból. Végül az Eclipse környezetben végrehajtott validálás sikeres volt.

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
3   <xs:element name="ételrendelések">
4     <xs:complexType>
5       <xs:sequence>
6         <xs:element name="rendelés" type="rendelésTipus" maxOccurs="unbounded"/>
7         <xs:element name="megrendelő" type="megrendelőTipus" maxOccurs="unbounded"/>
8         <xs:element name="fizetés" type="fizetésTipus" maxOccurs="unbounded"/>
9         <xs:element name="rendelt" type="rendeltTipus" maxOccurs="unbounded"/>
10        <xs:element name="étel" type="ételTipus" maxOccurs="unbounded"/>
11        <xs:element name="akció" type="akcióTipus" maxOccurs="unbounded"/>
12      </xs:sequence>
13    </xs:complexType>
14
15    <!-- Primary keys -->
16    <xs:key name="rendelésKulcs">
17      <xs:selector xpath="rendelés"/>
18      <xs:field xpath="@RID"/>
19    </xs:key>
20
21    <xs:key name="megrendelőKulcs">
22      <xs:selector xpath="megrendelő"/>
23      <xs:field xpath="@MID"/>
24    </xs:key>
25
26    <xs:key name="fizetésKulcs">
27      <xs:selector xpath="fizetés"/>
28      <xs:field xpath="@FID"/>
29    </xs:key>
30
31  </xs:element>
</xs:schema>
```

```

32<xs:key name="ételKulcs">
33    <xs:selector xpath="étel"/>
34    <xs:field xpath="@EID"/>
35</xs:key>
36
37<xs:key name="akcióKulcs">
38    <xs:selector xpath="akció"/>
39    <xs:field xpath="@AID"/>
40</xs:key>
41
42<!-- Foreign keys -->
43<xs:keyref refer="megrendelőKulcs" name="rendelésMegrendelőIdegenKulcs">
44    <xs:selector xpath="rendelés"/>
45    <xs:field xpath="@MID"/>
46</xs:keyref>
47
48<xs:keyref refer="rendelésKulcs" name="rendeltRendelésIdegenKulcs">
49    <xs:selector xpath="rendelt"/>
50    <xs:field xpath="@RID"/>
51</xs:keyref>
52
53<xs:keyref refer="ételKulcs" name="rendeltÉtelIdegenKulcs">
54    <xs:selector xpath="rendelt"/>
55    <xs:field xpath="@EID"/>
56</xs:keyref>
57
58<xs:keyref refer="ételKulcs" name="akcióÉtelIdegenKulcs">
59    <xs:selector xpath="akció"/>
60    <xs:field xpath="@EID"/>
61</xs:keyref>
62
63<!-- 1-1 -->
64<xs:unique name="rendelésFizetésEgy">
65    <xs:selector xpath="rendelés"/>
66    <xs:field xpath="@FID"/>
67</xs:unique>
68</xs:element>
69
70<!-- Saját típusok -->
71<xs:complexType name="rendelésTípus">
72    <xs:sequence>
73        <xs:element name="dátum" type="xs:date"/>
74        <xs:element name="kiszállítási_cím" type="xs:string"/>
75        <xs:element name="összeg" type="xs:integer"/>
76    </xs:sequence>
77    <xs:attribute name="RID" type="xs:integer" use="required"/>
78    <xs:attribute name="MID" type="xs:integer" use="required"/>
79    <xs:attribute name="FID" type="xs:integer" use="required"/>
80</xs:complexType>
81
82<xs:complexType name="megrendelőTípus">
83    <xs:sequence>
84        <xs:element name="visszatérő_vendég">
85            <xs:simpleType>
86                <xs:restriction base="xs:string">
87                    <xs:enumeration value="igen"/>
88                    <xs:enumeration value="nem"/>
89                </xs:restriction>
90            </xs:simpleType>
91        </xs:element>
92        <xs:element name="telefonszám" type="xs:string"/>
93        <xs:element name="név" type="xs:string"/>
94    </xs:sequence>
95    <xs:attribute name="MID" type="xs:integer" use="required"/>
96</xs:complexType>

```

```

97
98<xs:complexType name="fizetésTipus">
99  <xs:sequence>
100    <xs:element name="átvitel_ideje" type="xs:date"/>
101    <xs:element name="kupon">
102      <xs:simpleType>
103        <xs:restriction base="xs:string">
104          <xs:enumeration value="igen"/>
105          <xs:enumeration value="nem"/>
106        </xs:restriction>
107      </xs:simpleType>
108    </xs:element>
109    <xs:element name="fizetés_módja">
110      <xs:simpleType>
111        <xs:restriction base="xs:string">
112          <xs:enumeration value="bankkártya"/>
113          <xs:enumeration value="készpénz"/>
114          <xs:enumeration value="szép-kártya"/>
115        </xs:restriction>
116      </xs:simpleType>
117    </xs:element>
118  </xs:sequence>
119  <xs:attribute name="FID" type="xs:integer" use="required"/>
120</xs:complexType>
121
122<xs:complexType name="rendeltTipus">
123  <xs:sequence>
124    <xs:element name="mennyiség" type="xs:integer"/>
125  </xs:sequence>
126  <xs:attribute name="RID" type="xs:integer" use="required"/>
127  <xs:attribute name="EID" type="xs:integer" use="required"/>
128</xs:complexType>
129
130<xs:complexType name="ételTipus">
131  <xs:sequence>
132    <xs:element name="név" type="xs:string"/>
133    <xs:element name="ár" type="xs:integer"/>
134    <xs:element name="allergén" type="xs:string" maxOccurs="unbounded"/>
135    <xs:element name="típus" type="xs:string"/>
136  </xs:sequence>
137  <xs:attribute name="EID" type="xs:integer" use="required"/>
138</xs:complexType>
139
140<xs:complexType name="akcióTipus">
141  <xs:sequence>
142    <xs:element name="időtartam">
143      <xs:complexType>
144        <xs:sequence>
145          <xs:element name="kezdetek" type="xs:date"/>
146          <xs:element name="vége" type="xs:date"/>
147        </xs:sequence>
148      </xs:complexType>
149    </xs:element>
150    <xs:element name="leárazás" type="xs:integer"/>
151  </xs:sequence>
152  <xs:attribute name="EID" type="xs:integer" use="required"/>
153  <xs:attribute name="AID" type="xs:integer" use="required"/>
154</xs:complexType>
155
156</xs:schema>

```

2. feladat

A feladat egy DOM program készítése az XML dokumentum (XMLNeptunkod.xml) adatai adminisztrálása alapján.

2a) Adatolvasás – DOMRead

A külső „XMLF58KQ8.xml” fájlból beolvassuk az ételrendelés adatbázishoz tartozó adatokat. Minden egyedet külön feldolgozó metódus kezel, és írja ki a konzolra. Példa a konzolon kiírt eredményekre:

```
1 package hu.domparse.f58kq8;
2
3 import java.io.File;
4 import java.io.IOException;
5
6 import javax.xml.parsers.DocumentBuilder;
7 import javax.xml.parsers.DocumentBuilderFactory;
8 import javax.xml.parsers.ParserConfigurationException;
9 import javax.xml.transform.Transformer;
10 import javax.xml.transform.TransformerException;
11 import javax.xml.transform.TransformerFactory;
12 import javax.xml.transform.dom.DOMSource;
13 import javax.xml.transform.stream.StreamResult;
14
15 import org.w3c.dom.Document;
16 import org.w3c.dom.Element;
17 import org.w3c.dom.Node;
18 import org.w3c.dom.NodeList;
19 import org.xml.sax.SAXException;
20
21 public class DomReadF58KQ8 {
22
23     public static void main(String[] args) throws SAXException, IOException, ParserConfigurationException, TransformerException {
24         //xml fájl megnyitása ahonnan beolvassuk az adatokat
25         File xmlFile = new File("XMLF58KQ8.xml");
26
27         //dokumentum létrehozása
28         DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
29         DocumentBuilder dBuilder = factory.newDocumentBuilder();
30
31         //dokumentum létrehozása a beolvasott fájlból, majd annak a normalizálásáa
32         Document document = dBuilder.parse(xmlFile);
33         document.getDocumentElement().normalize();
34
35         System.out.println("Root element: " + document.getDocumentElement().getNodeName());
36
37         //beolvasó metódusok meghívása
38         getRendlés(document);
39
40         getMegrendelő(document);
41
42         getFizetés(document);
43
44         getKapcsoló(document);
45
46         getÉtelek(document);
47
48         getAkció(document);
49
50         //a módosított adatok kiírása txt fájlba fájlba
51         File modFile = new File("XMLF58KQ8.txt");
52
53         TransformerFactory transformerFactory = TransformerFactory.newInstance();
54         Transformer transformer = transformerFactory.newTransformer();
55
56         DOMSource source = new DOMSource(document);
57
58         StreamResult resultModFile = new StreamResult(modFile);
59         transformer.transform(source, resultModFile );
60
61     }
62 }
63
```

```

64 public static void getRendlés(Document doc) {
65     NodeList nList = doc.getElementsByTagName("rendelés");
66
67     System.out.println("\nMegrendelések:\n-----");
68
69     for (int i = 0; i < nList.getLength(); i++) {
70
71         Node nNode = nList.item(i);
72
73         System.out.println("\nCurrent Element: " + nNode.getNodeName());
74
75         if (nNode.getNodeType() == Node.ELEMENT_NODE) {
76
77             Element elem = (Element) nNode;
78             String rid = elem.getAttribute("RID");
79             String mid = elem.getAttribute("MID");
80             String fid = elem.getAttribute("FID");
81
82             Node node1 = elem.getElementsByTagName("dátum").item(0);
83             String node1Text = node1.getTextContent();
84
85             Node node2 = elem.getElementsByTagName("kiszállítási_cím").item(0);
86             String node2Text = node2.getTextContent();
87
88             Node node3 = elem.getElementsByTagName("összeg").item(0);
89             String node3Text = node3.getTextContent();
90
91             System.out.println("Rendelés id: " + rid);
92             System.out.println("Megrendelő id: " + mid);
93             System.out.println("Fizetés id: " + fid);
94             System.out.println("Dátum: " + node1Text);
95             System.out.println("Kiszállítási cím: " + node2Text);
96             System.out.println("Összeg: " + node3Text);
97         }
98     }
99 }
100
101
102 public static void getMegrendelő(Document doc) {
103     NodeList nList = doc.getElementsByTagName("megrendelő");
104
105     System.out.println("\nMegrendelők:\n-----");
106
107     for (int i = 0; i < nList.getLength(); i++) {
108
109         Node nNode = nList.item(i);
110
111         System.out.println("\nCurrent Element: " + nNode.getNodeName());
112
113         if (nNode.getNodeType() == Node.ELEMENT_NODE) {
114
115             Element elem = (Element) nNode;
116             String mid = elem.getAttribute("MID");
117
118             Node node1 = elem.getElementsByTagName("visszatérő_vendég").item(0);
119             String node1Text = node1.getTextContent();
120
121             Node node2 = elem.getElementsByTagName("telefonszám").item(0);
122             String node2Text = node2.getTextContent();
123
124             Node node3 = elem.getElementsByTagName("név").item(0);
125             String node3Text = node3.getTextContent();
126
127             System.out.println("Megrendelő id: " + mid);
128             System.out.println("Visszatérő vendég: " + node1Text);
129             System.out.println("Telefonszám: " + node2Text);
130             System.out.println("Név: " + node3Text);
131         }
132     }
133 }
134
135

```

```

136 public static void getFizetés(Document doc) {
137     NodeList nList = doc.getElementsByTagName("fizetés");
138
139     System.out.println("\nFizetések:\n-----");
140
141     for (int i = 0; i < nList.getLength(); i++) {
142
143         Node nNode = nList.item(i);
144
145         System.out.println("\nCurrent Element: " + nNode.getNodeName());
146
147         if (nNode.getNodeType() == Node.ELEMENT_NODE) {
148
149             Element elem = (Element) nNode;
150             String fid = elem.getAttribute("FID");
151
152             Node node1 = elem.getElementsByTagName("átvitel_ideje").item(0);
153             String node1Text = node1.getTextContent();
154
155             Node node2 = elem.getElementsByTagName("kupon").item(0);
156             String node2Text = node2.getTextContent();
157
158             Node node3 = elem.getElementsByTagName("fizetés_módja").item(0);
159             String node3Text = node3.getTextContent();
160
161             System.out.println("Fizetés id: " + fid);
162             System.out.println("Átvitel ideje: " + node1Text);
163             System.out.println("Kupon: " + node2Text);
164             System.out.println("Fizetés módja: " + node3Text);
165         }
166     }
167 }
168
169
170 public static void getKapcsoló(Document doc) {
171     NodeList nList = doc.getElementsByTagName("rendelt");
172
173     System.out.println("\nMegrendelt ételek kapcsolói:\n-----");
174
175     for (int i = 0; i < nList.getLength(); i++) {
176
177         Node nNode = nList.item(i);
178
179         System.out.println("\nCurrent Element: " + nNode.getNodeName());
180
181         if (nNode.getNodeType() == Node.ELEMENT_NODE) {
182
183             Element elem = (Element) nNode;
184             String rid = elem.getAttribute("RID");
185             String eid = elem.getAttribute("EID");
186
187             Node node1 = elem.getElementsByTagName("mennyiség").item(0);
188             String node1Text = node1.getTextContent();
189
190             System.out.println("Rendelés id: " + rid);
191             System.out.println("Étel id: " + eid);
192             System.out.println("Mennyiség: " + node1Text);
193         }
194     }
195 }
196
197

```

```

198 public static void getÉtelek(Document doc) {
199     NodeList nList = doc.getElementsByTagName("étel");
200
201     System.out.println("\nÉtelek:\n-----");
202
203     for (int i = 0; i < nList.getLength(); i++) {
204
205         Node nNode = nList.item(i);
206
207         System.out.println("\nCurrent Element: " + nNode.getNodeName());
208
209         if (nNode.getNodeType() == Node.ELEMENT_NODE) {
210
211             Element elem = (Element) nNode;
212             String eid = elem.getAttribute("EID");
213
214             Node node1 = elem.getElementsByTagName("név").item(0);
215             String node1Text = node1.getTextContent();
216
217             Node node2 = elem.getElementsByTagName("ár").item(0);
218             String node2Text = node2.getTextContent();
219
220
221
222
223
224             System.out.println("Ételek id: " + eid);
225             System.out.println("Név: " + node1Text);
226             System.out.println("Ár: " + node2Text);
227
228             int counter = 0;
229
230             while (elem.getElementsByTagName("allergén").item(counter) != null) {
231                 Node node3 = elem.getElementsByTagName("allergén").item(counter);
232                 String node3Text = node3.getTextContent();
233                 System.out.println("Allergén: " + node3Text);
234                 counter++;
235             }
236
237
238             Node node4 = elem.getElementsByTagName("típus").item(0);
239             String node4Text = node4.getTextContent();
240
241             System.out.println("Típus: " + node4Text);
242         }
243     }
244 }
245
246
247 public static void getAkció(Document doc) {
248     NodeList nList = doc.getElementsByTagName("akció");
249
250     System.out.println("\nAkciók:\n-----");
251
252     for (int i = 0; i < nList.getLength(); i++) {
253
254         Node nNode = nList.item(i);
255
256         System.out.println("\nCurrent Element: " + nNode.getNodeName());
257
258         if (nNode.getNodeType() == Node.ELEMENT_NODE) {
259
260             Element elem = (Element) nNode;
261             String aid = elem.getAttribute("AID");
262             String eid = elem.getAttribute("EID");
263
264             Node node1 = elem.getElementsByTagName("kezdet").item(0);
265             String node1Text = node1.getTextContent();
266
267             Node node2 = elem.getElementsByTagName("vége").item(0);
268             String node2Text = node2.getTextContent();
269
270             Node node3 = elem.getElementsByTagName("leírás").item(0);
271             String node3Text = node3.getTextContent();
272
273             System.out.println("Akció id: " + aid);
274             System.out.println("Étel id: " + eid);
275             System.out.println("Kezdet: " + node1Text);
276             System.out.println("Vége: " + node2Text);
277             System.out.println("Leírás: " + node3Text);
278         }
279     }
280 }
281 }
282 }
283

```

2b) Adatmódosítás – DOMModify

A beolvasott xml fájlban 5 különböző módosítás végrehajtása. Majd azok kiírása egy új fájlba, nem változtatva az eredeti xml fájlt.

Az öt módosítás a következő:

- a 'szép-kártyával' történő fizetések átírása 'bankkártyára'
- a harmadik megrendelés megrendelőjét a harmadik megrendelőre változtatjuk, az első megrendelő helyett
- a 10% leárazású akciók növelése 20-ra
- a Piedone nevű pizza árának csökkentése 500 forintra
- azok a megrendelők, akik eddig nem voltak visszatérő vendégek, azokat azzá módosítani

```
1 package hu.domparse.f58kq8;
2
3 import java.io.File;
4
5 import javax.xml.parsers.DocumentBuilder;
6 import javax.xml.parsers.DocumentBuilderFactory;
7 import javax.xml.transform.Transformer;
8 import javax.xml.transform.TransformerFactory;
9 import javax.xml.transform.dom.DOMSource;
10 import javax.xml.transform.stream.StreamResult;
11
12 import org.w3c.dom.Document;
13 import org.w3c.dom.NamedNodeMap;
14 import org.w3c.dom.Node;
15 import org.w3c.dom.NodeList;
16
17 public class DomModifyF58KQ8 {
18
19     public static void main(String[] args) {
20         try {
21             //xml fájl megnyitása ahonnan beolvassuk az adatokat
22             File file = new File("XMLF58KQ8.xml");
23
24             //dokumentum létrehozása a beolvasott fájlból
25             DocumentBuilder documentBuilder = DocumentBuilderFactory.newInstance().newDocumentBuilder();
26             Document document = documentBuilder.parse(file);
27
28             //dokumentum normalizálása
29             document.getDocumentElement().normalize();
```



```

30
31
32 //1. módosítás - a 'szép-kártyával' történő fizetések átírása 'bankkártyára':
33 NodeList fizetesNList = document.getElementsByTagName("fizetés");
34 for (int temp = 0; temp < fizetesNList.getLength(); temp++) {
35     Node node = fizetesNList.item(temp);
36
37     if (node.getNodeType() == Node.ELEMENT_NODE) {
38
39         NodeList childNodes = node.getChildNodes();
40
41         for (int j = 0; j < childNodes.getLength(); j++) {
42
43             Node childNode = childNodes.item(j);
44
45             if ("fizetés_módja".equals(childNode.getNodeName())) {
46
47                 if ("szép-kártya".equals(childNode.getTextContent())) {
48                     childNode.setTextContent("bankkártya");
49                 }
50             }
51         }
52     }
53 }
54
55 //2. módosítás - a harmadik megrendelés megrendelőjét a harmadik megrendelőre változtatjuk, az első megrendelő helyett:
56 NodeList rendelNList = document.getElementsByTagName("rendelés");
57 for (int temp = 0; temp < rendelNList.getLength(); temp++) {
58     Node node = rendelNList.item(temp);
59
60     NamedNodeMap attr = node.getAttributes();
61     Node nodeAttrM = attr.getNamedItem("MID");
62     Node nodeAttrR = attr.getNamedItem("RID");
63
64     if (nodeAttrR.getTextContent().equals("3")) {
65         nodeAttrM.setTextContent("3");
66     }
67 }
68
69 //3. módosítás - a 10% leárazású akciók növelése 20-ra
70 NodeList akcioNList = document.getElementsByTagName("akció");
71 for (int temp = 0; temp < akcioNList.getLength(); temp++) {
72     Node node = akcioNList.item(temp);
73
74     if (node.getNodeType() == Node.ELEMENT_NODE) {
75
76         NodeList childNodes = node.getChildNodes();
77
78         for (int j = 0; j < childNodes.getLength(); j++) {
79
80             Node childNode = childNodes.item(j);
81
82             if ("leárazás".equals(childNode.getNodeName())) {
83
84                 if ("10".equals(childNode.getTextContent())) {
85                     childNode.setTextContent("20");
86                 }
87             }
88         }
89     }
90 }
91

```

```

92 //4. módosítás - a Piedone nevű pizza árának csökkentése 500 forinttal
93 NodeList etelNList = document.getElementsByTagName("étel");
94 for (int temp = 0; temp < etelNList.getLength(); temp++) {
95     Node node = etelNList.item(temp);
96
97     if (node.getNodeType() == Node.ELEMENT_NODE) {
98
99         NodeList childNodes = node.getChildNodes();
100
101         for (int j = 0; j < childNodes.getLength(); j++) {
102             Node childNode = childNodes.item(j);
103
104             if ("név".equals(childNode.getNodeName())) {
105                 if ("Piedone".equals(childNode.getTextContent())) {
106
107                     for (int k = 0; k < childNodes.getLength(); k++) {
108                         childNode = childNodes.item(k);
109                         if ("ár".equals(childNode.getNodeName())) {
110                             childNode.setTextContent("1750");
111                         }
112                     }
113                 }
114             }
115         }
116     }
117 }
118
119 //5. módosítás - azok a megrendelők, akik eddig nem voltak visszatérő vendégek, azokat azzá módosítani
120 NodeList megrendeloNList = document.getElementsByTagName("megrendelő");
121 for (int temp = 0; temp < megrendeloNList.getLength(); temp++) {
122     Node node = megrendeloNList.item(temp);
123
124     if (node.getNodeType() == Node.ELEMENT_NODE) {
125
126         NodeList childNodes = node.getChildNodes();
127
128         for (int j = 0; j < childNodes.getLength(); j++) {
129             Node childNode = childNodes.item(j);
130
131             if ("visszatérő_vendég".equals(childNode.getNodeName())) {
132
133                 if ("nem".equals(childNode.getTextContent())) {
134                     childNode.setTextContent("igen");
135                 }
136             }
137         }
138     }
139 }
140
141 //a módosított adatok kiírása mind konzolra mind fájlba
142 File modFile = new File("MOD_XMLF58KQ8.xml");
143
144 TransformerFactory transformerFactory = TransformerFactory.newInstance();
145 Transformer transformer = transformerFactory.newTransformer();
146
147 DOMSource source = new DOMSource(document);
148
149 System.out.println("---- Módosított fajl -----");
150
151 StreamResult consoleResult = new StreamResult(System.out);
152 StreamResult resultModFile = new StreamResult(modFile);
153
154 transformer.transform(source, consoleResult);
155 transformer.transform(source, resultModFile);
156
157 } catch (Exception e) {
158     e.printStackTrace();
159 }
160 }
161 }
162 }

```

Végül pedig egy txt formátumú fájlba kerülnek az xml fájl adatai.

Root element: ételrendelések

Megrendelések:

Current Element: rendelés

Rendelés id: 1

Megrendelő id: 1

Fizetés id: 1

Dátum: 2021-01-01

Kiszállítási cím: Arany_János_utca_1.

Összeg: 6750

Current Element: rendelés

Rendelés id: 2

Megrendelő id: 2

Fizetés id: 2

Dátum: 2022-02-02

Kiszállítási cím: Nagy_Lasjos_király_útja_2.

Összeg: 6060

Current Element: rendelés

Rendelés id: 3

Megrendelő id: 1

Fizetés id: 3

Dátum: 2021-10-10

Kiszállítási cím: Arany_János_utca_1.

Összeg: 10880

Current Element: rendelés

Rendelés id: 4

Megrendelő id: 3

Fizetés id: 4

Dátum: 2022-11-04

Kiszállítási cím: Kuruc_utca_23.

Összeg: 9470

2c) Adatlekérdezés – DOMQuery

Az xml fájlból beolvasott adatok meghatározott lekérdezése. A lekérdezések gyakran egyéb feltételekhez kötöttek, ezek az attribútumok vagy tulajdonságok értékeihez vannak kötve. A lekérdezések eredményeit futás után kiíratjuk a konzolra.

A következő 5 lekérdezés van megvalósítva a programban:

- a 'Húsétel' típusú ételek nevének és árának lekérdezése
- azoknak a rendeléseknek lekérdezése, amelyek az 1. ID-jű rendelőtől jöttek, 10000ft alatti összeg értékben
- hány fizetés történt a különböző fizetési módok használatával
- a NEM visszatérő vendégek lekérdezése
- a 6 ID-jű ételhez tartozó akciók lekérdezése

```
1 package hu.domparse.f58kq8;
2
3 import java.io.File;
4 import java.io.IOException;
5
6 import javax.xml.parsers.DocumentBuilder;
7 import javax.xml.parsers.DocumentBuilderFactory;
8 import javax.xml.parsers.ParserConfigurationException;
9
10 import org.w3c.dom.Document;
11 import org.w3c.dom.Element;
12 import org.w3c.dom.NamedNodeMap;
13 import org.w3c.dom.Node;
14 import org.w3c.dom.NodeList;
15 import org.xml.sax.SAXException;
16
17 public class DomQueryF58KQ8 {
18
19     public static void main(String[] args) throws IOException, ParserConfigurationException, SAXException {
20         try {
21             //xml fájl megnyitása ahonnan beolvassuk az adatokat
22             File file = new File("XMLF58KQ8.xml");
23
24             //dokumentum létrehozása létrehozása a beolvasott fájlból
25             DocumentBuilder documentBuilder = DocumentBuilderFactory.newInstance().newDocumentBuilder();
26             Document document = documentBuilder.parse(file);
27
28             //dokumentum normalizálása
29             document.getDocumentElement().normalize();
30         }
```

```

32 //1. lekérdezés: a 'Húsétel' típusú ételek nevének és árának lekérdezése
33 System.out.println("\n1. a 'Húsétel' típusú ételek nevének és árának lekérdezése:");
34 NodeList nList1 = document.getElementsByTagName("étel");
35
36 for (int i = 0; i < nList1.getLength(); i++) {
37     Node nNode = nList1.item(i);
38
39     if (nNode.getNodeType() == Node.ELEMENT_NODE) {
40
41         Element elem = (Element) nNode;
42         NodeList childNodes = nNode.getChildNodes();
43
44         for (int j = 0; j < childNodes.getLength(); j++) {
45             Node childNode = childNodes.item(j);
46
47             if ("típus".equals(childNode.getNodeName())) {
48
49                 if ("Húsétel".equals(childNode.getTextContent())) {
50                     System.out.println("");
51
52                     String eid = elem.getAttribute("EID");
53                     Node node1 = elem.getElementsByTagName("név").item(0);
54                     String node1Text = node1.getTextContent();
55                     Node node2 = elem.getElementsByTagName("ár").item(0);
56                     String node2Text = node2.getTextContent();
57
58                     System.out.println("Ételek id: " + eid);
59                     System.out.println("Név: " + node1Text);
60                     System.out.println("Ár: " + node2Text);
61
62                 }
63             }
64         }
65     }
66 }
67
68 System.out.println("-----");
69
70
71
72
73 //2. lekérdezés: azoknak a rendeléseknek lekérdezése amelyek az 1. ID-jú rendelőtől jöttek, 10000ft alatti összeg értékben
74 System.out.println("\n2. lekérdezés: azoknak a rendeléseknek lekérdezése, amelyek az 1. ID-jú rendelőtől jöttek, 10000ft alatti összeg értékben:");
75 NodeList nList2 = document.getElementsByTagName("rendelés");
76
77 for (int i = 0; i < nList2.getLength(); i++) {
78     Node nNode = nList2.item(i);
79
80     if (nNode.getNodeType() == Node.ELEMENT_NODE) {
81
82         Element elem = (Element) nNode;
83         NodeList childNodes = nNode.getChildNodes();
84
85         NamedNodeMap attr = nNode.getAttributes();
86         Node nodeAttrM = attr.getNamedItem("MID");
87
88         if (nodeAttrM.getTextContent().equals("1")) {
89             for (int j = 0; j < childNodes.getLength(); j++) {
90                 Node childNode = childNodes.item(j);
91
92                 if ("összeg".equals(childNode.getNodeName())) {
93
94                     if (Integer.valueOf(childNode.getTextContent()) < 10000 ) {
95                         System.out.println("");
96
97                         String rid = elem.getAttribute("RID");
98                         Node node1 = elem.getElementsByTagName("datum").item(0);
99                         String node1Text = node1.getTextContent();
100                         Node node2 = elem.getElementsByTagName("kiszállítási cím").item(0);
101                         String node2Text = node2.getTextContent();
102                         Node node3 = elem.getElementsByTagName("összeg").item(0);
103                         String node3Text = node3.getTextContent();
104
105                         System.out.println("Rendelés id: " + rid);
106                         System.out.println("Datum: " + node1Text);
107                         System.out.println("Kiszállítási cím: " + node2Text);
108                         System.out.println("Összeg: " + node3Text);
109
110                     }
111                 }
112             }
113         }
114     }
115 }
116
117 System.out.println("-----");
118
119
120

```

```

123 //3. lekérdezés: hány fizetés történt a különböző fizetési módok használatával
124 System.out.println("\n3. hány fizetés történt a különböző fizetési módok használatával");
125 NodeList nList3 = document.getElementsByTagName("fizetés");
126
127 int bankSzám = 0;
128 int szépSzám = 0;
129 int pénzSzám = 0;
130
131 for (int i = 0; i < nList3.getLength(); i++) {
132
133     Node nNode = nList3.item(i);
134
135     if (nNode.getNodeType() == Node.ELEMENT_NODE) {
136
137         NodeList childNodes = nNode.getChildNodes();
138
139         for (int j = 0; j < childNodes.getLength(); j++) {
140
141             Node childNode = childNodes.item(j);
142
143             if ("fizetés_módja".equals(childNode.getNodeName())) {
144
145                 if ("bankkártya".equals(childNode.getTextContent())) {
146                     bankSzám++;
147                 }
148                 else if ("szép-kártya".equals(childNode.getTextContent())) {
149                     szépSzám++;
150                 }
151                 else if ("késpénz".equals(childNode.getTextContent())) {
152                     pénzSzám++;
153                 }
154             }
155         }
156     }
157 }
158
159 System.out.println("\nBankkártyás fizetések száma: " + bankSzám);
160 System.out.println("Szép-kártyás fizetések száma: " + szépSzám);
161 System.out.println("Késpénzes fizetések száma: " + pénzSzám);
162 System.out.println("-----");
163
164
165 //4. lekérdezés: a NEM visszatérő vendégek lekérdezése
166 System.out.println("\n4. a NEM visszatérő vendégek lekérdezése");
167 NodeList nList4 = document.getElementsByTagName("megrendelő");
168
169 for (int i = 0; i < nList4.getLength(); i++) {
170
171     Node nNode = nList4.item(i);
172
173     if (nNode.getNodeType() == Node.ELEMENT_NODE) {
174
175         Element elem = (Element) nNode;
176         NodeList childNodes = nNode.getChildNodes();
177
178         for (int j = 0; j < childNodes.getLength(); j++) {
179
180             Node childNode = childNodes.item(j);
181
182             if ("visszatérő_vendég".equals(childNode.getNodeName())) {
183
184                 if ("nem".equals(childNode.getTextContent())) {
185                     System.out.println("");
186
187                     String mid = elem.getAttribute("MID");
188                     Node node1 = elem.getElementsByTagName("név").item(0);
189                     String node1Text = node1.getTextContent();
190
191                     System.out.println("Megrendelő id: " + mid);
192                     System.out.println("Név: " + node1Text);
193                 }
194             }
195         }
196     }
197 }
198
199 System.out.println("-----");
200

```

```

201 //5. lekérdezés: a 6 ID-jű ételhez tartozó akciók lekérdezése
202 System.out.println("\n5. a 6 ID-jű ételhez tartozó akciók lekérdezése:");
203 NodeList nList5 = document.getElementsByTagName("akció");
204
205 for (int i = 0; i < nList5.getLength(); i++) {
206     Node nNode = nList5.item(i);
207
208     if (nNode.getNodeType() == Node.ELEMENT_NODE) {
209         Element elem = (Element) nNode;
210         NamedNodeMap attr = nNode.getAttributes();
211         Node nodeAttrM = attr.getNamedItem("EID");
212
213         if (nodeAttrM.getTextContent().equals("6")) {
214
215             System.out.println("");
216
217             String aid = elem.getAttribute("AID");
218             Node node1 = elem.getElementsByTagName("kezdet").item(0);
219             String node1Text = node1.getTextContent();
220             Node node2 = elem.getElementsByTagName("vége").item(0);
221             String node2Text = node2.getTextContent();
222             Node node3 = elem.getElementsByTagName("leárazás").item(0);
223             String node3Text = node3.getTextContent();
224
225             System.out.println("Akció id: " + aid);
226             System.out.println("Akció kezdete: " + node1Text);
227             System.out.println("Akció vége: " + node2Text);
228             System.out.println("Leárazás mértéke: " + node3Text);
229         }
230     }
231 }
232
233 System.out.println("-----");
234
235
236
237
238
239
240 } catch (IOException e) {
241     e.printStackTrace();
242 } catch (ParserConfigurationException e) {
243     e.printStackTrace();
244 } catch (SAXException e) {
245     e.printStackTrace();
246 }
247
248 }
249 }

```

A lekérdezések eredményei konzolra kiírva:

1. a 'Húsétel' típusú ételek nevének és árának lekérdezése:

Ételek id: 3
Név: Szezámós_csirke_box
Ár: 2720

Ételek id: 4
Név: Gyros_tál
Ár: 2720

2. lekérdezés: azoknak a rendeléseknek lekérdezése, amelyek az 1. ID-jű rendelőtől jöttek, 10000ft alatti összeg értékben:

Rendelés id: 1
Dátum: 2021-01-01
Kiszállítási cím: Arany_János_utca_1.
Összeg: 6750

3. hány fizetés történt a különböző fizetési módok használatával:

Bankkártyás fizetések száma: 1
Szép-kártyás fizetések száma: 1
Készpénzes fizetések száma: 2

4. a NEM visszatérő vendégek lekérdezése:

Megrendelő id: 2
Név: Második_Mária

Megrendelő id: 3
Név: Kis_Pista

5. a 6 ID-jű ételhez tartozó akciók lekérdezése:

Akció id: 4
Akció kezdete: 2022-04-15
Akció vége: 2022-05-15
Leárazás mértéke: 15

Akció id: 5
Akció kezdete: 2022-09-01
Akció vége: 2022-11-30
Leárazás mértéke: 30