

# JEGYZŐKÖNYV

Adatkezelés XML környezetben  
Féléves feladat

Készítette: **Bajusz István**

Neptunkód: **OBYBZK**

## A feladat leírása:

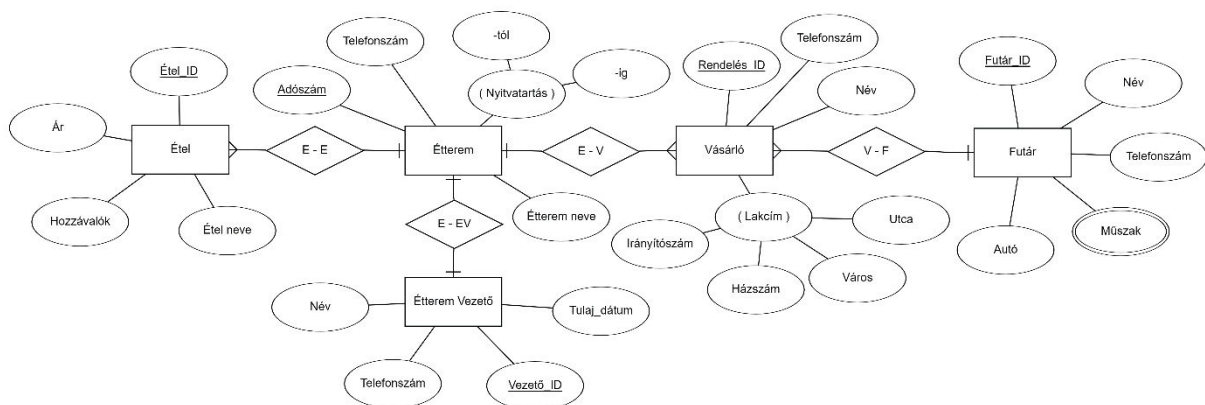
A feladatom bemutatja az éttermek fontosabb adatainak tárolását.

Az ER modell mutatja az étteremben lévő adatok egymáshoz való kapcsolódását. Az egyedeket egy-egy egyedi tulajdonságokkal láttam el (ID). Az első kapcsolat az étterem és az étel között van. Ez egy N:1 kapcsolat, mivel egy étteremhez tartozhat többféle étel is, viszont egy ételt egyszerre csak egy étteremhez köthetünk. Az ételek az „Étel\_ID” nevű egyedi tulajdonsággal, illetve az „Ár”, „Hozzávalók” és az „Étel neve” tulajdonságokkal rendelkeznek. Az ételeket az étterem készíti, amelynek van neve, telefonszáma, nyitvatartási ideje (-tól, -ig), illetve adószáma (ID). Az étteremből lehetőség van ételrendelésre, így a következő kapcsolat a „Vásárló” és az „Étterem” egyed között jött létre. Ez egy N:M kapcsolat, mivel egy étteremnek lehet több megrendelője is, illetve egy megrendelő rendelhet több étteremből is. A megrendelőnek van lakcíme (irányítószám, város, utca, házszám), telefonszáma, neve és rendelés ID-je. Az utolsó egyed a „Futár”. A futár és a vásárló közötti kapcsolat a kiszállítás, ami szintén egy N:M kapcsolat, mivel egy vásárlóhoz tartozhat egyszerre több futár is (több helyről is rendelhet), illetve egy futárnak lehet egyszerre több címe is, ahova ki kell szállítania. Egy futárnak van Futár ID-je, neve, telefonszáma, autója, illetve egy műszakot tároló tulajdonság. Utóbbi többször is előfordulhat egy futárnál. Az étteremnek van egy vezetője. Ebben az esetben egy vezetőnek csak egy étterme lehet, így a két egyed között 1:1 kapcsolat jön létre. A vezetőnek van neve, telefonszáma, Vezető\_ID-je, illetve egy tulajDátum tulajdonsága (saját típus), amit azt jelzi, hogy mióta ő az étterem vezetője.

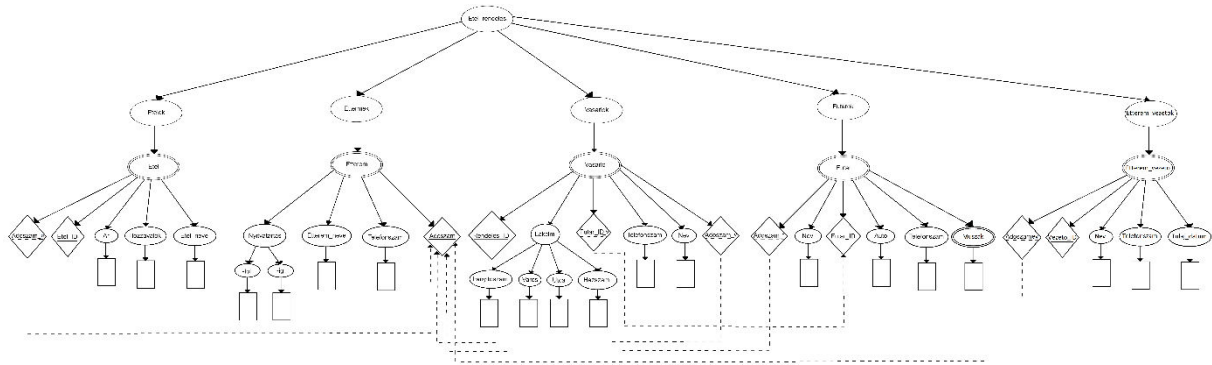
A rendeléseket a Vásárló tábla fogja rögzíteni, az elérhető összes ételt az Étel tábla, az elérhető éttermeket az Étterem tábla, illetve a dolgozó futárokat a Futár tábla. A vásárló idegen kulcsa fogja tárolni, hogy melyik étteremből rendelt. A futár szintén idegen kulccsal köthető egy étteremhez.

## 1. feladat

### 1a) Az adatbázis ER modell:



### 1b) Az adatbázis konvertálása XDM modellre:



### 1c) Az XDM modell alapján XML dokumentum készítése:

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

```
<!-- etelek -->
```

```
<etel Adoszam_e="123456789101" Etel_ID="01">
  <etel_neve>hamburger</etel_neve>
  <ar>1590</ar>
  <hozzavalok>buci, huspogacs, paradicsom, szoszok, salata</hozzavalok>
</etel>
<etel Adoszam_e="123456789102" Etel_ID="02">
  <etel_neve>pizza</etel_neve>
  <ar>2390</ar>
  <hozzavalok>paradicsomos alap, sonka, gomba, kukorica, sajt</hozzavalok>
</etel>
<etel Adoszam_e="123456789103" Etel_ID="03">
  <etel_neve>gyros</etel_neve>
  <ar>2190</ar>
  <hozzavalok>gyros hus, zoldsegek, hasaburgonya, szosz</hozzavalok>
</etel>
```

&lt;!-- etelek vege --&gt;

```
<!-- ettermek -->
```

```
<etterem Adoszam="123456789101">  
  <etterem_neve>Első etterem</etterem_neve>  
  <telefonszam>06301234123</telefonszam>  
  <nyitvatartas>
```

```
<tol>8</tol>
<ig>18</ig>
</nyitvatartas>
</etterem>
<etterem Adoszam="123456789102">
  <etterem_neve>Masodik etterem</etterem_neve>
  <telefonszam>06201234123</telefonszam>
  <nyitvatartas>
    <tol>8</tol>
    <ig>22</ig>
  </nyitvatartas>
</etterem>
<etterem Adoszam="123456789103">
  <etterem_neve>Harmadik etterem</etterem_neve>
  <telefonszam>06701234123</telefonszam>
  <nyitvatartas>
    <tol>10</tol>
    <ig>18</ig>
  </nyitvatartas>
</etterem>

<!-- ettermek vege -->
<!-- vasarlok -->

<vasarlo Adoszam_v="123456789101" Futar_ID_v="11" Rendeles_ID="1">
  <nev>Nagy Bela</nev>
  <telefonszam>06301234567</telefonszam>
  <lakcim>
    <iranyitoszam>3600</iranyitoszam>
    <varos>Ozd</varos>
    <utca>Kossuth</utca>
    <hazszam>12</hazszam>
  </lakcim>
</vasarlo>
<vasarlo Adoszam_v="123456789102" Futar_ID_v="12" Rendeles_ID="2">
  <nev>Kovacs Andrea</nev>
  <telefonszam>06701234567</telefonszam>
  <lakcim>
    <iranyitoszam>2220</iranyitoszam>
    <varos>Vecses</varos>
    <utca>Fo</utca>
    <hazszam>3</hazszam>
  </lakcim>
</vasarlo>
<vasarlo Adoszam_v="123456789103" Futar_ID_v="13" Rendeles_ID="3">
  <nev>Kiss Sandor</nev>
  <telefonszam>06201234567</telefonszam>
```

```
<lakcim>
  <iranyitoszam>1098</iranyitoszam>
  <varos>Budapest</varos>
  <utca>Petofi</utca>
  <hazszam>42</hazszam>
</lakcim>
</vasarlo>

<!-- vasarlok vege -->
<!-- futarok -->

<futar Adoszam_f="123456789101" Futar_ID="11">
  <nev>Kiss Bela</nev>
  <auto>suzuki</auto>
  <telefonszam>06309876543</telefonszam>
  <muszak>Delelott</muszak>
</futar>
<futar Adoszam_f="123456789102" Futar_ID="12">
  <nev>Kovacs Ferenc</nev>
  <auto>toyota</auto>
  <telefonszam>06309876543</telefonszam>
  <muszak>Delelott</muszak>
  <muszak>Delutan</muszak>
</futar>
<futar Adoszam_f="123456789103" Futar_ID="13">
  <nev>Nemeth Sandor</nev>
  <auto>opel</auto>
  <telefonszam>06309876543</telefonszam>
  <muszak>Delutan</muszak>
</futar>

<!-- futarok vege -->
<!-- etteremVezetok -->

<etteremVezeto Vezeto_ID="001" Adoszamev="123456789101">
  <nev>Vezeto Bela</nev>
  <telefonszam>06704567891</telefonszam>
  <tulajDatum>2011</tulajDatum>
</etteremVezeto>
<etteremVezeto Vezeto_ID="002" Adoszamev="123456789102">
  <nev>Kovacs Andrea</nev>
  <telefonszam>06304567891</telefonszam>
  <tulajDatum>2008</tulajDatum>
</etteremVezeto>
<etteremVezeto Vezeto_ID="003" Adoszamev="123456789103">
  <nev>Nagy Ferenc</nev>
  <telefonszam>06204567891</telefonszam>
```

```

        <tulajDatum>2021</tulajDatum>
    </etteremVezeto>

    <!-- etteremVezetok vege -->

</etel_rendeles>

```

## 1d) Az XML dokumentum alapján XML Schema készítése (saját típus):

```

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

        <!-- Az XML dokumentum felepítése -->

        <xs:complexType>
            <xs:sequence>
                <xs:element name="etel" type="etelTipus" maxOccurs="unbounded" />
                <xs:element name="etterem" type="etteremTipus" maxOccurs="unbounded" />
                <xs:element name="vasarlo" type="vasarloTipus" maxOccurs="unbounded" />
                <xs:element name="futar" type="futarTipus" maxOccurs="unbounded" />
                <xs:element name="etteremVezeto" type="etteremVezetoTipus"
maxOccurs="unbounded" />
            </xs:sequence>
        </xs:complexType>

        <!-- Kulcsok -->

        <xs:key name="etel_kulcs">
            <xs:selector xpath="etel" />
            <xs:field xpath="@Etel_ID"></xs:field>
        </xs:key>
        <xs:key name="etterem_kulcs">
            <xs:selector xpath="etterem" />
            <xs:field xpath="@Adoszam"></xs:field>
        </xs:key>
        <xs:key name="vasarlo_kulcs">
            <xs:selector xpath="vasarlo" />
            <xs:field xpath="@Rendeles_ID"></xs:field>
        </xs:key>
        <xs:key name="futar_kulcs">
            <xs:selector xpath="futar" />
            <xs:field xpath="@Futar_ID"></xs:field>

```

```

</xs:key>
<xs:key name="etteremVezeto_kulcs">
  <xs:selector xpath="etteremVezeto"></xs:selector>
  <xs:field xpath="@Vezeto_ID"></xs:field>
</xs:key>

<!-- Idegen kulcsok -->

<xs:keyref refer="etterem_kulcs" name="etel_etterem_idegen_kulcs">
  <xs:selector xpath="etel"></xs:selector>
  <xs:field xpath="@Adoszam_e"></xs:field>
</xs:keyref>
<xs:keyref refer="futar_kulcs" name="vasarlo_futar_idegen_kulcs">
  <xs:selector xpath="vasarlo"></xs:selector>
  <xs:field xpath="@Futar_ID_v"></xs:field>
</xs:keyref>
<xs:keyref refer="etterem_kulcs" name="vasarlo_etterem_idegen_kulcs">
  <xs:selector xpath="vasarlo"></xs:selector>
  <xs:field xpath="@Adoszam_v"></xs:field>
</xs:keyref>
<xs:keyref refer="etterem_kulcs" name="futar_etterem_idegen_kulcs">
  <xs:selector xpath="futar"></xs:selector>
  <xs:field xpath="@Adoszam_f"></xs:field>
</xs:keyref>
<xs:keyref refer="etterem_kulcs" name="etteremVezeto_etterem_idegen_kulcs">
  <xs:selector xpath="etteremVezeto"></xs:selector>
  <xs:field xpath="@Adoszamev"></xs:field>
</xs:keyref>

<xs:unique name="unique_etteremVezeto">
  <xs:selector xpath="etteremVezeto"></xs:selector>
  <xs:field xpath="@Adoszamev"></xs:field>
</xs:unique>
</xs:element>

<!-- Tipusok -->
<!-- etel tipus -->

<xs:complexType name="etelTipus">
  <xs:sequence>
    <xs:element type="xs:string" name="etel_neve"/>
    <xs:element type="xs:short" name="ar"/>
    <xs:element type="xs:string" name="hozzavalok"/>
  </xs:sequence>
  <xs:attribute type="xs:integer" name="Etel_ID" use="required"/>
  <xs:attribute type="xs:integer" name="Adoszam_e" use="required"/>
</xs:complexType>

```

<!-- etterem tipus -->

```
<xs:complexType name="etteremTipus">
  <xs:sequence>
    <xs:element type="xs:string" name="etterem_neve"/>
    <xs:element type="xs:string" name="telefonszam"/>
    <xs:element name="nyitvatartas">
      <xs:complexType>
        <xs:sequence>
          <xs:element type="xs:short" name="tol"/>
          <xs:element type="xs:short" name="ig"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute type="xs:integer" name="Adoszam" use="required"/>
</xs:complexType>
```

<!-- vasarlo tipus -->

```
<xs:complexType name="vasarloTipus">
  <xs:sequence>
    <xs:element type="xs:string" name="nev"/>
    <xs:element type="xs:string" name="telefonszam"/>
    <xs:element name="lakcim">
      <xs:complexType>
        <xs:sequence>
          <xs:element type="xs:string" name="iranyitoszam"/>
          <xs:element type="xs:string" name="varos"/>
          <xs:element type="xs:string" name="utca"/>
          <xs:element type="xs:string" name="hazszam"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute type="xs:integer" name="Rendeles_ID" use="required"/>
  <xs:attribute type="xs:integer" name="Adoszam_v" use="required"/>
  <xs:attribute type="xs:integer" name="Futar_ID_v" use="required"/>
</xs:complexType>
```

<!-- futar tipus -->

```
<xs:complexType name="futarTipus">
  <xs:sequence>
    <xs:element type="xs:string" name="nev"/>
    <xs:element type="xs:string" name="auto"/>
```



```

        <xs:element type="xs:string" name="telefonszam"/>
        <xs:element type="xs:string" name="muszak" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute type="xs:integer" name="Futar_ID" use="required"/>
    <xs:attribute type="xs:integer" name="Adoszam_f" use="required"/>
</xs:complexType>

<!-- etteremVezeto tipus -->

<xs:complexType name="etteremVezetoTipus">
    <xs:sequence>
        <xs:element type="xs:string" name="nev"/>
        <xs:element type="xs:string" name="telefonszam"/>
        <xs:element name="tulajDatum">
            <xs:simpleType>
                <xs:restriction base="xs:gYear">
                    <xs:minInclusive value="1950"/>
                    <xs:maxInclusive value="2021"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
    </xs:sequence>
    <xs:attribute type="xs:integer" name="Vezeto_ID" use="required"/>
    <xs:attribute type="xs:integer" name="Adoszamev" use="required"/>
</xs:complexType>

</xs:schema>

```

## 2. feladat

### 2a) Adatolvasás - DOMReadObybzk.java

A program kiolvassa az XML dokumentum tartalmát, majd kimenetként megadja ezeket. A program futtatásakor lefut a DOMModifyObybzk.java osztály is, ami megnöveli az ételek árát 200-zal. Az adatmódosítás után ismét kiírja az adatokat a program, jelezve, hogy változtatás történt.

```

1 package hu.domparsed.obybkz;
2
3 import java.io.File;
4
5 public class DOMReadObybkz {
6
7     private static void printNode(Node r) {
8
9         // Node nevenek kiirása
10        if (r.getNodeName() != "#text") {
11            System.out.println(r.getNodeName());
12        }
13
14        // Gyerekek node listája helvarasa
15        NodeList children = r.getChildNodes();
16        for (int i = 0; i < children.getLength(); i++) {
17            Node child = children.item(i);
18            boolean isComplex = child.getTextContent().contains("\n");
19
20            // Attribute kiirása
21            if (child.hasAttributes()) {
22                NamedNodeMap attributes = child.getAttributes();
23                int numAttrs = attributes.getLength();
24                for (int j = 0; j < numAttrs; j++) {
25                    Attr attr = (Attr) attributes.item(j);
26                    String attrName = attr.getNodeName();
27                    String attrValue = attr.getNodeValue();
28                    System.out.println(" " + attrName + " : " + attrValue);
29                }
30            }
31
32            // Nev es tartalom kiirása
33            if (isComplex) {
34                printNode(child);
35            } else {
36                System.out.print(" " + child.getNodeName());
37                System.out.println(": " + child.getTextContent());
38            }
39        }
40    }
41
42    public static void main(String[] args) {
43        try {
44
45            DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
46            DocumentBuilder dbBuilder = dbFactory.newDocumentBuilder();
47            Document doc = dbBuilder.parse("src/hu/domparsed/obybkz/XMLObybkz.xml");
48            String filepath = "src/hu/domparsed/obybkz/XMLObybkz.xml";
49            doc.getDocumentElement().normalize();
50
51            // XPath
52            XPath xPath = XPathFactory.newInstance().newXPath();
53
54            Element root = doc.getDocumentElement();
55
56            System.out.println("\n ----- \n Modositás előtt: \n ----- \n");
57
58            printNode(root);
59            DOMModifyObybkz.modifyPrices(root);
60
61            System.out.println("\n ----- \n Modositás után: \n ----- \n");
62
63            printNode(root);
64
65            // Modosított XML mentése
66            TransformerFactory transformerFactory = TransformerFactory.newInstance();
67            Transformer transformer = transformerFactory.newTransformer();
68            DOMSource source = new DOMSource(doc);
69            StreamResult result = new StreamResult(new File(filepath));
70            transformer.transform(source, result);
71
72        } catch (Exception e) {
73            e.printStackTrace();
74        }
75    }
76 }

```

## 2b) Adatmódosítás – DOMModifyObybzk.java

A program megnöveli az ételek árát 200-zal.

```
1 package hu.domparse.obybkz;
2
3 import org.w3c.dom.Node;
4
5
6 public class DOMModifyObybzk {
7
8     public static void modifyPrices(Node root) {
9         try {
10
11             System.out.println(root.getNodeType());
12             NodeList els = root.getChildNodes();
13
14             // Vegigmegy a root (etel_rendeles) gyerekein
15             for (int j = 0; j < els.getLength(); j++) {
16
17                 // Ha a gyerek neve etel, akkor végrehajtja a következőket
18                 if (els.item(j).getNodeName() == "etel") {
19
20                     // Vegigmegy a gyerek elemein
21                     for (int k = 0; k < els.item(j).getChildNodes().getLength(); k++) {
22
23                         // Ha az egyik elem neve ar, akkor megnöveli annak értékét 200-al
24                         if (els.item(j).getChildNodes().item(k).getNodeName() == "ar") {
25
26                             int ar = Integer.parseInt(els.item(j).getChildNodes().item(k).getTextContent());
27                             els.item(j).getChildNodes().item(k).setTextContent(String.valueOf(ar + 200));
28
29                         }
30                     }
31                 }
32             }
33
34         } catch (Exception e) {
35             e.printStackTrace();
36         }
37     }
38 }
```

## 2c) Adatlekérdezés – DOMQueryObybzk.java

A program kimenetként megadja azoknak a futároknak a nevét, akik suzuki vezetnek, illetve azoknak az éttermeknek az adószámát, akik árulnak gyrost. Ha nem található olyan futár, aki suzuki vezet, vagy olyan étterem, amelyik gyrost árul, akkor mindkét esetben ezt jelzi a kimenetben.

```

1 package hu.domparse.obybkz;
2
3 import java.io.File;
4
5
6 public class DOMQueryObybkz {
7
8     public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException {
9
10         File xmlFile = new File("src/hu/domparse/obybkz/XML0bybkz.xml");
11
12         DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
13         DocumentBuilder dBuilder = factory.newDocumentBuilder();
14
15         Document doc = dBuilder.parse(xmlFile);
16
17         doc.getDocumentElement().normalize();
18
19         System.out.println("Root: " + doc.getDocumentElement().getNodeName() + "\n");
20
21         // Kiírja azokat a futarokat, akiknek suzuki autójuk van
22         System.out.println("A suzuki-t hasznalo futarok neve: \n");
23
24         NodeList nodeList = doc.getElementsByTagName("futar");
25
26         boolean foundAny = false;
27
28         for(int i = 0; i < nodeList.getLength(); i++) {
29
30             Node node = nodeList.item(i);
31
32             if(node.getNodeType() == Node.ELEMENT_NODE) {
33                 Element elem = (Element) node;
34
35                 Node node2;
36
37                 // Megvizsgálom az autója típusát
38                 node2 = elem.getElementsByTagName("auto").item(0);
39                 String carType = node2.getTextContent();
40
41                 // Ha suzuki, akkor kiírom a futar nevét
42                 if("suzuki".equals(carType)) {
43                     Node nodeName = elem.getElementsByTagName("nev").item(0);
44                     String name = nodeName.getTextContent();
45
46                     System.out.println(name + "\n");
47
48                     foundAny = true;
49                 }
50             }
51
52             if(foundAny == false) {
53
54                 System.out.println("\n Nem található olyan futar, aki suzuki-t vezet. \n");
55             }
56             else {
57                 foundAny = false;
58             }
59
60             // Kiírja azoknak az ettermek a nevét, ahol kapható gyros
61             System.out.println("\n Gyrost arulo ettermek: \n");
62
63             nodeList = doc.getElementsByTagName("etel");
64
65             foundAny = false;
66
67             for(int i = 0; i < nodeList.getLength(); i++) {
68
69                 Node node = nodeList.item(i);
70
71                 if(node.getNodeType() == Node.ELEMENT_NODE) {
72                     Element elem = (Element) node;
73
74                     Node node2;
75
76                     // Megvizsgálja az ételét
77                     node2 = elem.getElementsByTagName("etel_neve").item(0);
78                     String foodName = node2.getTextContent();
79
80                     // Ha gyros, akkor kiírja az ételt aulin étterem adószámát
81                     if("gyros".equals(foodName)) {
82
83                         String taxNumber = elem.getAttribute("Adoszam_e");
84
85                         System.out.println(taxNumber + "\n");
86
87                         foundAny = true;
88                     }
89                 }
90
91             }
92
93             if(foundAny == false) {
94
95                 System.out.println("\n Nem található olyan étterem az adatbázisban, amelyik arul gyrost.\n");
96             }
97
98         }
99
100     }
101 }

```