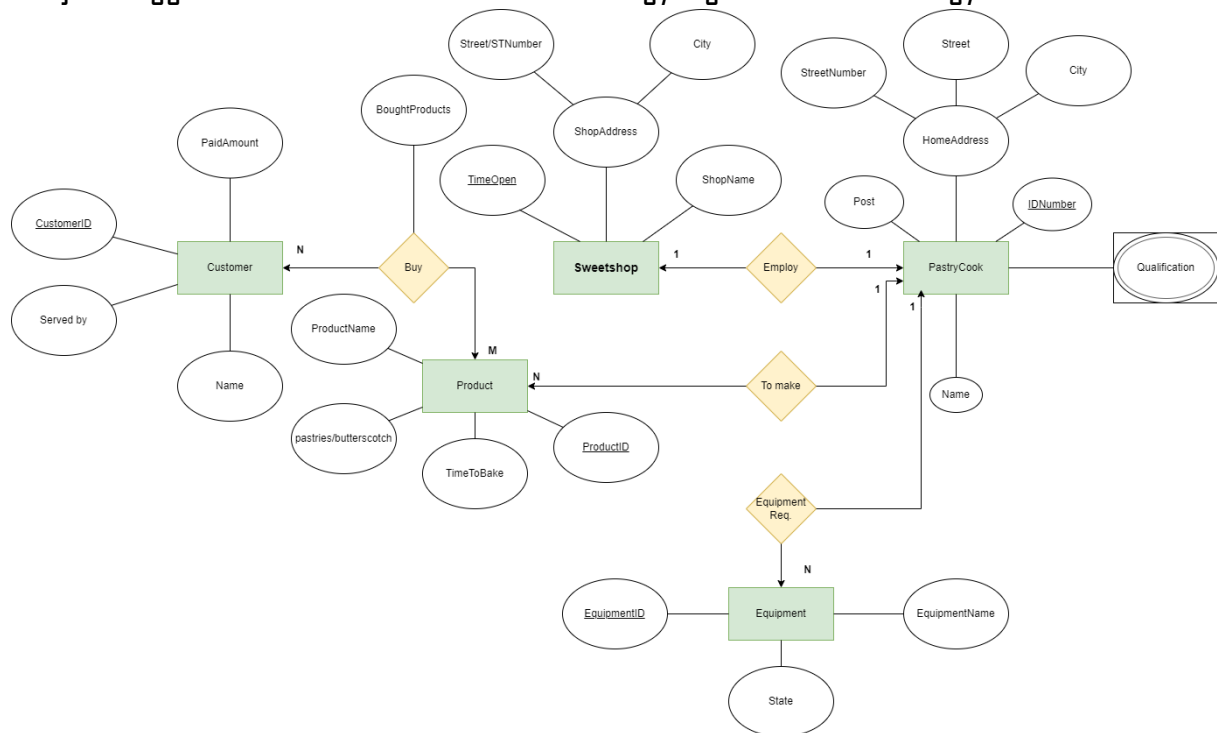


Jegyzőkönyv
Adatkezelés XML
környezetben
Féléves Feladat

Kádár Konrád
IWWD97

A feladat leírása:

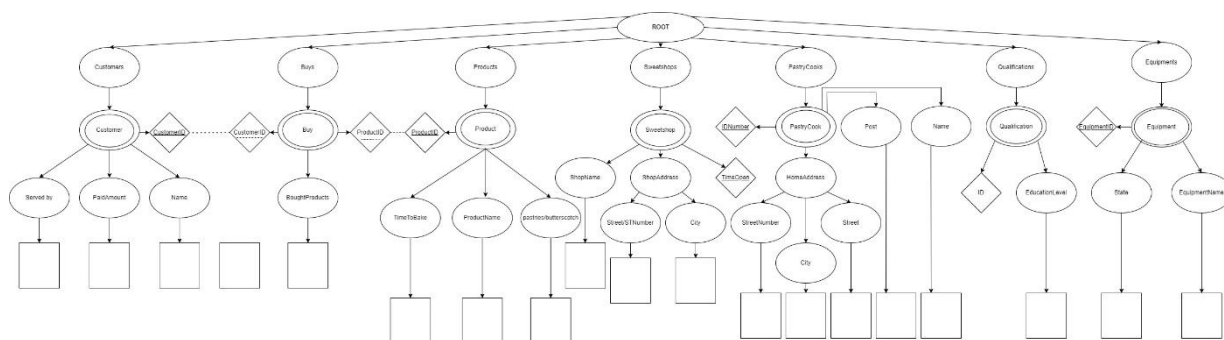
A beadandó témája egy cukrász üzlet csoport és annak adatbázis rendszere. Az egységek között megtaláljuk a vásárlókat, dolgozókat, cukrászdákat, termékeket és azok elkészítéséhez szükséges eszközöket. A „Sweetshop” és a „PastryCook” egységek összetett tulajdonsággal is rendelkeznek. Minden egység rendelkezik egyedi azonosítóval.



XDM modell:

ER modellt 3 jelölés segítségével konvertálhatunk XDM modellre.

Van az elem, amely az XDM modellben ellipszissel van rajzolva. Van az attribútum, jelölése rombusz, szöveges tartalmat téglalappal jelöljük. Ezeken kívül, ha egy elem többszörös előfordulási lehetőséggel bír azt dupla ellipszissel jelöljük. A többi a többhöz kapcsolatokat külön meg kell mutatni.



XML és XSD:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <root xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="XMLSchemaTUD077.xsd">
3   <Customers>
4     <Customer CustomerID="1">
5       <ServedBy>Minta Cukrász</ServedBy>
6       <PaidAmount>3000</PaidAmount>
7       <Name>Minta Vassarló</Name>
8     </Customer>
9   </Customers>
10  <Products>
11    <Product ProductID="1" PastryCookKey2="1">
12      <ProductName>Iorta</ProductName>
13      <PastriesButterscotch>Butterscotch</PastriesButterscotch>
14      <TimeToBake>2</TimeToBake>
15    </Product>
16    <Product ProductID="11" PastryCookKey2="2">
17      <ProductName>Sajtos Tallér</ProductName>
18      <PastriesButterscotch>Pastry</PastriesButterscotch>
19      <TimeToBake>1</TimeToBake>
20    </Product>
21  </Products>
22  <Buys>
23    <Buy CustomerID="1" ProductID="1">
24      <BoughtProducts>Édes Sütemény</BoughtProducts>
25    </Buy>
26  </Buys>
27  <Sweetshops>
28    <Sweetshop TimeOpen="1">
29      <ShopName>Cukrászda</ShopName>
30      <ShopAddress>
31        <City>Debrecen</City>
32        <StreetNumber>Hátyás király utca 4</StreetNumber>
33      </ShopAddress>
34    </Sweetshop>
35    <Sweetshop TimeOpen="2">
36      <ShopName>Cukrászda2</ShopName>
37      <ShopAddress>
38        <City>Debrecen</City>
39        <StreetNumber>Sásóni út 68</StreetNumber>
40      </ShopAddress>
41    </Sweetshop>
42  </Sweetshops>
43 </root>
44 </pre>
45 <?xml version="1.0" encoding="UTF-8"?>
46 <xsi:schema xmlns:xsi="http://www.w3.org/2001/XMLSchema">
47   <xsi:element name="root">
48     <xsi:complexType>
49       <xsi:sequence>
50         <xsi:element name="Customers">
51           <xsi:complexType>
52             <xsi:sequence>
53               <xsi:element name="Customer" maxOccurs="100">
54                 <xsi:complexType>
55                   <xsi:sequence>
56                     <xsi:element name="ServedBy" type="xs:string"/>
57                     <xsi:element name="PaidAmount" type="xs:float"/>
58                     <xsi:element name="Name" type="xs:string"/>
59                   </xsi:sequence>
60                 </xsi:complexType>
61               </xsi:element>
62             </xsi:sequence>
63           </xsi:complexType>
64         </xsi:element>
65         <xsi:element name="Products">
66           <xsi:complexType>
67             <xsi:sequence>
68               <xsi:element name="Product" maxOccurs="unbounded">
69                 <xsi:complexType>
70                   <xsi:sequence>
71                     <xsi:element name="ProductName" type="xs:string"/>
72                     <xsi:element name="PastriesButterscotch" type="xs:string"/>
73                     <xsi:element name="TimeToBake" type="xs:int"/>
74                   </xsi:sequence>
75                   <xsi:attribute name="ProductID" type="xs:int"/>
76                   <xsi:attribute name="PastryCookKey2" type="xs:int"/>
77                 </xsi:complexType>
78               </xsi:element>
79             </xsi:sequence>
80           </xsi:complexType>
81         </xsi:element>
82         <xsi:element name="Buys">
83           <xsi:complexType>
84             <xsi:sequence>
85               <xsi:element name="Buy">
86                 <xsi:complexType>
87                   <xsi:sequence>
88                     <xsi:element name="CustomerID" type="xs:int"/>
89                     <xsi:element name="ProductID" type="xs:int"/>
90                     <xsi:element name="BoughtProducts" type="xs:string"/>
91                   </xsi:sequence>
92                 </xsi:complexType>
93               </xsi:element>
94             </xsi:sequence>
95           </xsi:complexType>
96         </xsi:element>
97         <xsi:element name="Sweetshops">
98           <xsi:complexType>
99             <xsi:sequence>
100              <xsi:element name="Sweetshop" maxOccurs="unbounded">
101                <xsi:complexType>
102                  <xsi:sequence>
103                    <xsi:element name="TimeOpen" type="xs:int"/>
104                    <xsi:element name="ShopName" type="xs:string"/>
105                    <xsi:element name="ShopAddress">
106                      <xsi:complexType>
107                        <xsi:sequence>
108                          <xsi:element name="City" type="xs:string"/>
109                          <xsi:element name="StreetNumber" type="xs:string"/>
110                        </xsi:sequence>
111                      </xsi:complexType>
112                    </xsi:sequence>
113                  </xsi:complexType>
114                </xsi:element>
115              </xsi:sequence>
116            </xsi:complexType>
117          </xsi:element>
118        </xsi:sequence>
119      </xsi:complexType>
120    </xsi:element>
121  </xsi:schema>
```

```
XML2 XMLSchemaWWS07.xsd
XMLSchemaWWS07.xsd
33 22 17
</Sweetshops>
<PastryCook IDNumber="527738EE" SweetshopID="1">
  <PostVezetőCukrász/Post>
    <Name>Minta Cukrász</Name>
    <HomeAddress>
      <City>Debrecen</City>
      <Street>Leningen utca</Street>
      <StreetNumber>45</StreetNumber>
    </HomeAddress>
  </PastryCook>
  <PastryCook IDNumber="527638EE" SweetshopID="1">
    <PostVezetőCukrász/Post>
      <Name>Minta Cukrász</Name>
      <HomeAddress>
        <City>Debrecen</City>
        <Street>Alma utca</Street>
        <StreetNumber>34</StreetNumber>
      </HomeAddress>
    </PastryCook>
  </PastryCook>
  <Qualifications>
    <Qualification ID="2">
      <EducationLevel>Szakunkas</EducationLevel>
    </Qualification>
    <Qualification ID="1">
      <EducationLevel>Érettség</EducationLevel>
    </Qualification>
  </Qualifications>
  <Equipments>
    <Equipment ID="1" PastryCookID="1">
      <State>Használatra Kész</State>
      <EquipmentName>Kész</EquipmentName>
    </Equipment>
    <Equipment ID="2" PastryCookID="1">
      <State>Használatra Kész</State>
      <EquipmentName>Tepsi</EquipmentName>
    </Equipment>
  </Equipments>
</root>
root - Bugs
<xs:sequence>
  <xs:element name="Buy" maxOccurs="10">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="BoughtProducts" type="xs:string"/>
      </xs:sequence>
      <xs:attribute name="CustomerID" type="xs:int"/>
      <xs:attribute name="ProductID" type="xs:int"/>
    </xs:complexType>
  </xs:element>
</xs:sequence>
<xs:element name="Sweetshops">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Sweetshop" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ShopName" type="xs:string"/>
            <xs:element name="ShopAddress">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="City" type="xs:string"/>
                  <xs:element name="StreetsNumber" type="xs:string"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
          <xs:attribute name="lineOpen" type="xs:int"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="PastryCooks">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="PastryCook" maxOccurs="5">
        <xs:complexType>
          <xs:sequence>

```

```
XML2 XMLSchemaWWS07.xsd
XMLSchemaWWS07.xsd
22 17
<xs:selector xpath="Equipment"></xs:selector>
<xs:field xpath="ID"></xs:field>
</xs:keyref>
<xs:keyref name="ProductPastryCook" refer="PastryCookKey">
  <xs:selector xpath="Product"></xs:selector>
  <xs:field xpath="PastryCookKey"></xs:field>
</xs:keyref>
<xs:keyref name="ProductBuy" refer="ProductKey">
  <xs:selector xpath="Buy"></xs:selector>
  <xs:field xpath="ProductID"></xs:field>
</xs:keyref>
<xs:keyref name="BuyandCustomer" refer="CustomerKey">
  <xs:selector xpath="Buy"></xs:selector>
  <xs:field xpath="CustomerID"></xs:field>
</xs:keyref>
<xs:keyref name="SweetshopPastryCook" refer="SweetshopKey">
  <xs:selector xpath="Sweetshop"></xs:selector>
  <xs:field xpath="SweetshopID"></xs:field>
</xs:keyref>
</xs:element>
<xs:simpleType name="Fizetett">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="8000"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Cégcím">
  <xs:restriction base="xs:string">
    <xs:maxLength value="50"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Szemlé">
  <xs:restriction base="xs:string">
    <xs:pattern value="[0-9]{6}[A-Z]{2}"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>
xs:schema xs:element xs:complexType xs:sequence xs:element
2241 LF UTF-8 4 spaces
```

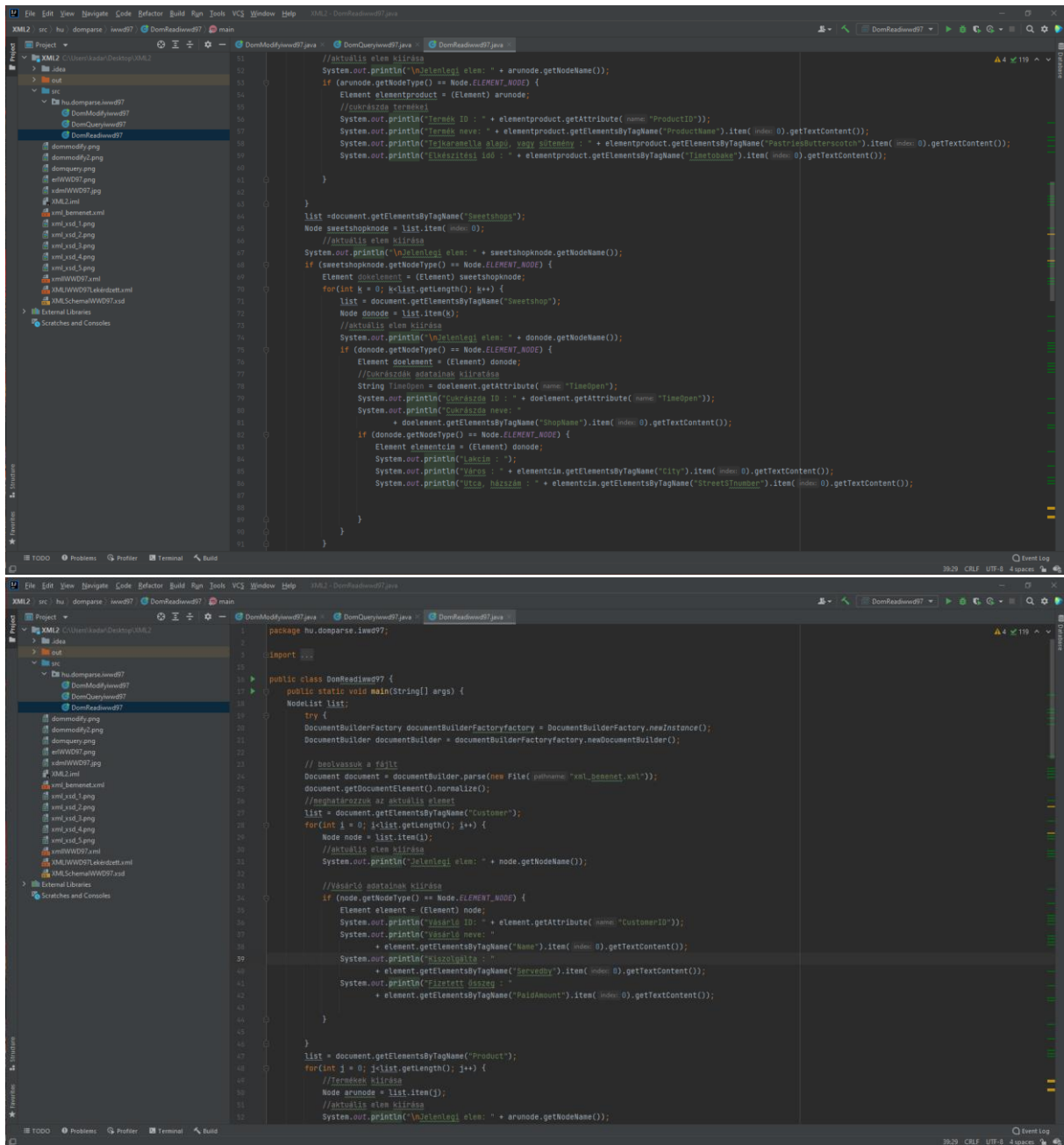


```

1 package hu.domparsa.iwv97;
2
3 import org.w3c.dom.*;
4
5 public class DomModifyV97 {
6     public static void main(String[] args) {
7         NodeList list;
8
9         try {
10             DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
11             DocumentBuilder builder = factory.newDocumentBuilder();
12
13             // fáj1.html
14             Document document = builder.parse(new File("path\\xml_beszam.xml"));
15             document.getDocumentElement().normalize();
16
17             // Aktubilis elem megkeresése
18             list = document.getElementsByTagName("Customer");
19
20             for (int i = 0; i < list.getLength(); i++) {
21                 Node node = list.item(i);
22
23                 // Aktubilis elem kiírása
24                 System.out.println("Current element: " + node.getNodeName());
25
26                 if (node.getNodeType() == Node.ELEMENT_NODE) {
27                     Element element = (Element) node;
28                     //customerID elcsúszás egy írással
29                     String customercode=element.getAttribute( "name:CustomerID");
30                     //újabb vásárló adatainak kiírása
31                     System.out.println("VasarId neve : "
32                                     + element.getElementsByTagName("name").item(0).getTextContent());
33                     System.out.println("VasarId ID : " + element.getAttribute( "name:CustomerID"));
34                     System.out.println("Pozitív Száma : "
35                                     + element.getElementsByTagName("PozitivSzam").item(0).getTextContent() );
36                     System.out.println("VasarIdja : "
37                                     + element.getElementsByTagName("VasarId").item(0).getTextContent() );
38
39                     Node customercode = list.item(i);
40
41                     if (customercode.getNodeType() == Node.ELEMENT_NODE) {
42                         Element element = (Element) node;
43                         //customercode elcsúsztatás 2-vel akkor, Nev módosítása Hosszok VasarId
44                         if (customercode.equals("1")) {
45                             element1.getElementsByTagName("Name").item(0).setTextContent("Hosszok VasarId");
46                         }
47                     }
48                     //Új vásárló adatainak kiírása
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

DomRead:



```
11 //aktuális elem kiírása
12 System.out.println("Aktuális elem: " + arunode.getNodeName());
13 if (arunode.getNodeType() == Node.ELEMENT_NODE) {
14     Element elementproduct = (Element) arunode;
15     //Cikkszám kiírása
16     System.out.println("Termék ID : " + elementproduct.getAttribute("name: ProductID"));
17     System.out.println("Termék neve: " + elementproduct.getElementsByTagName("ProductName").item(0).getTextContent());
18     System.out.println("Teljes ár: " + elementproduct.getElementsByTagName("Price").item(0).getTextContent());
19     System.out.println("Elérhető mennyiség: " + elementproduct.getElementsByTagName("Quantity").item(0).getTextContent());
20 }
21 }
22 list = document.getElementsByTagName("Sweetshop");
23 Node sweetshopnode = list.item(0);
24 //aktuális elem kiírása
25 System.out.println("Aktuális elem: " + sweetshopnode.getNodeName());
26 if (sweetshopnode.getNodeType() == Node.ELEMENT_NODE) {
27     Element doelement = (Element) sweetshopnode;
28     for (int k = 0; k < list.getLength(); k++) {
29         list = document.getElementsByTagName("Sweetshop");
30         Node donode = list.item(k);
31         //aktuális elem kiírása
32         System.out.println("Aktuális elem: " + donode.getNodeName());
33         if (donode.getNodeType() == Node.ELEMENT_NODE) {
34             Element doelement = (Element) donode;
35             //Cikkszám kiírása
36             String timeopen = doelement.getAttribute("name: TimeOpen");
37             System.out.println("Cikkszám ID : " + doelement.getAttribute("name: TimeOpen"));
38             System.out.println("Cikkszám neve: "
39                 + doelement.getElementsByTagName("ShopName").item(0).getTextContent());
40             if (doelement.getElementsByTagName("City").item(0).getLength() > 0) {
41                 Element elementcity = (Element) doelement;
42                 System.out.println("Létszám : ");
43                 System.out.println("Ár: " + elementcity.getElementsByTagName("City").item(0).getTextContent());
44                 System.out.println("Utca, házszám : " + elementcity.getElementsByTagName("StreetNumber").item(0).getTextContent());
45             }
46         }
47     }
48 }
49 }
50 }
51 }
```

```
1 package hu.domparse.lwe97;
2
3 import java.io.*;
4
5 public class DomReadme97 {
6     public static void main(String[] args) {
7         Node list;
8         try {
9             DocumentBuilderFactory documentBuilderFactory = DocumentBuilderFactory.newInstance();
10             DocumentBuilder documentBuilder = documentBuilderFactory.newDocumentBuilder();
11             // beolvasás a fájl
12             Document document = documentBuilder.parse(new File("xml_bemenet.xml"));
13             document.getDocumentElement().normalize();
14             //meghatározom az aktuális elem
15             list = document.getElementsByTagName("Customer");
16             for (int i = 0; i < list.getLength(); i++) {
17                 Node node = list.item(i);
18                 //aktuális elem kiírása
19                 System.out.println("Aktuális elem: " + node.getNodeName());
20                 //Várlék kiírása
21                 if (node.getNodeType() == Node.ELEMENT_NODE) {
22                     Element element = (Element) node;
23                     System.out.println("Várlék ID : " + element.getAttribute("name: CustomerID"));
24                     System.out.println("Várlék neve: "
25                         + element.getElementsByTagName("Name").item(0).getTextContent());
26                     System.out.println("Kiszámlázás : "
27                         + element.getElementsByTagName("Serviced").item(0).getTextContent());
28                     System.out.println("Fizetett összeg : "
29                         + element.getElementsByTagName("PaidAmount").item(0).getTextContent());
30                 }
31             }
32             list = document.getElementsByTagName("Product");
33             for (int j = 0; j < list.getLength(); j++) {
34                 //termék kiírása
35                 Node arunode = list.item(j);
36                 //aktuális elem kiírása
37                 System.out.println("Aktuális elem: " + arunode.getNodeName());
38             }
39         } catch (Exception e) {
40             e.printStackTrace();
41         }
42     }
43 }
```