

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Készítette: **Kozma János**

Neptun kód: **SVOXGH**

FELADAT LEÍRÁSA

Feladatomban egy cipőboltból való online rendelés folyamatát mutatom be, egészen a termék kiválasztásától a futárszolgálathoz való átadásig.

Az ER modell szemlélteti a cipőboltban található adatok egymáshoz való kapcsolódását. Az egyedek rendelkeznek egyedi azonosító tulajdonságokkal (ID). Cipő és Rendelés közötti kapcsolat a kosárhoz adást szemlélteti (C-R), amely egy N:1 kapcsolatot, hiszen több cipőt is lehet rendelni, viszont egy cipőt egyszerre csakis egy rendeléshez köthetünk.

A „Cipő” egyedek rendelkeznek egy egyedi tulajdonsággal „C_id”, valamint „Márka”, „Név”, „Méret”, „Ár”, „Készlet” tulajdonságokkal.

A rendelést a vásárlók adják le, ezt az R-V kapcsolatot szemlélteti, amely egy N:1 kapcsolatot, hiszen egy vásárló több rendelést is leadhat, viszont egy rendelés csakis egy vásárlóhoz tartozik.

A „Vásárló” egyed rendelkezik egy „V_id” egyedi tulajdonsággal, illetve „Név”, és „Cím” (Ir.szám, Település, Utca, Házszám) tulajdonságokkal.

A „Rendelés” egyed rendelkezik egy azonosítóval (R_id), valamint egy „Kedvezmény” tulajdonsággal. A rendelés és számla közötti kapcsolat egy 1:1 típusú, hiszen minden rendeléshez kizárólag egy számla kerül kiállításra.

A „Számla” egyed a „SZ_id” egyedi tulajdonság, valamint „Mennyiség”, „Összeg”, „Dátum” (saját típus) tulajdonságokkal rendelkezik.

Az utolsó kapcsolat a rendelés és a futárszolgálat közötti kapcsolat, amely a kézbesítésre átadást szemlélteti, ez egy N:M kapcsolatot, hiszen a rendeléseket több futárszolgálat is kézbesítheti, valamint a futárszolgálatokhoz több rendelés is beérkezhet.

A „Futárszolgálat” egyed a „Fsz_id” egyedi tulajdonsággal, valamint „Név” tulajdonsággal rendelkezik.

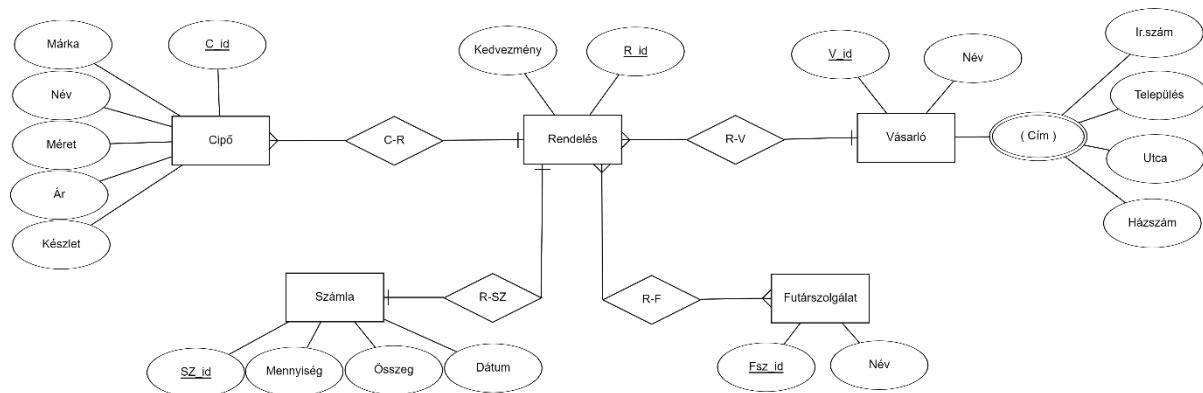
A rendeléseket a Rendelés tábla rögzíti, a rendelésben szereplő cipő(k) adatai és a rendelésről kiállított számla, valamint a vásárló is idegenkulccsal köthető a rendelésekhez.

A rendelés és a futárszolgálat szintén

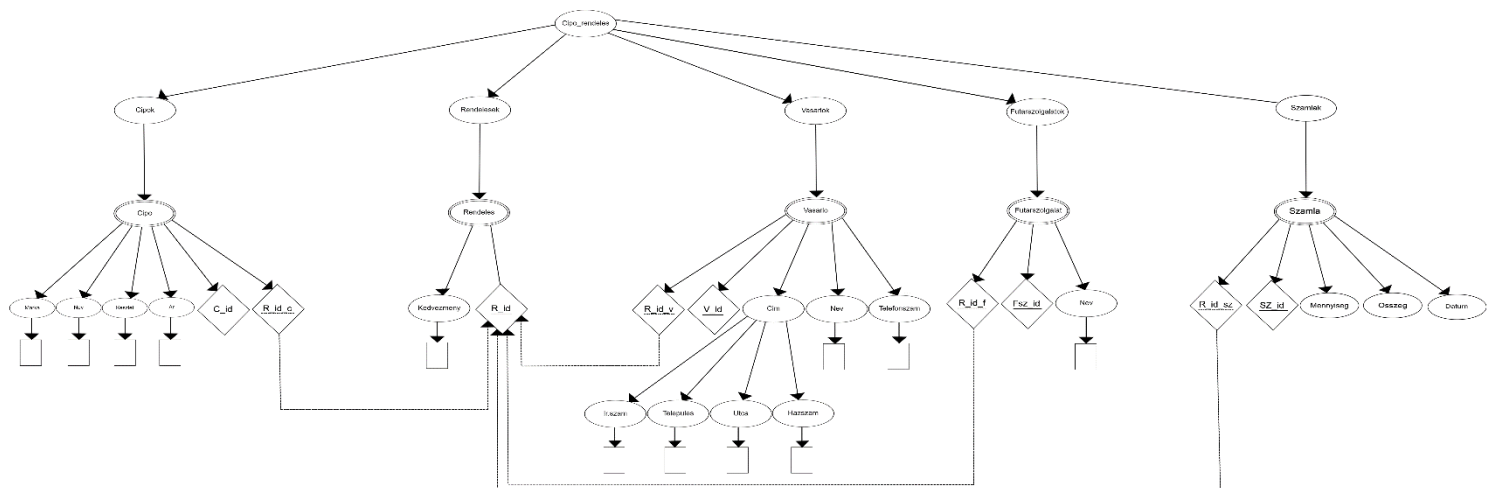
Az elérhető összes cipőt a Cipő tábla, a vásárlók adatait a Vásárló tábla, az elérhető futárszolgálatokat a Futárszolgálat tábla, illetve a számlák adatait a Számla tábla rögzíti.

1. FELADAT

a) Az adatbázis ER modell



b) Az adatbázis konvertálása XDM modellre



c) Az XDM modell alapján XML dokumentum készítése

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

  <!-- cipok -->

  <cipo R_id_c="123456789101" C_id="01">
    <marka>Nike</marka>
    <cipo_neve>Air Force</cipo_neve>
    <szin>Fehér</szin>
    <ar>26000</ar>
    <keszlet>200</keszlet>
  </cipo>
  <cipo R_id_c="123456789102" C_id="02">
    <marka>Adidas</marka>
    <cipo_neve>ZX 2K BOOST</cipo_neve>
    <szin>Piros</szin>
  </cipo>
</cipo_rendeles>
```

```

        <ar>22500</ar>
        <keszlet>100</keszlet>
    </cipo>
    <cipo R_id_c="123456789103" C_id="03">
        <marka>Vans</marka>
        <cipo_neve>UltraRange EXO Hi</cipo_neve>
        <szin>Fekete - fehér</szin>
        <ar>32000</ar>
        <keszlet>45</keszlet>
    </cipo>
    <cipo R_id_c="123456789104" C_id="04">
        <marka>Nike</marka>
        <cipo_neve>Air Jordan Retro Mid</cipo_neve>
        <szin>Fekete - sárga</szin>
        <ar>26000</ar>
        <keszlet>20</keszlet>
    </cipo>

    <!-- cipok vege -->
    <!-- rendelések -->

    <rendeles R_id="123456789101">
        <kedvezmeny>2000</kedvezmeny>
    </rendeles>
    <rendeles R_id="123456789102">
        <kedvezmeny>3000</kedvezmeny>
    </rendeles>
    <rendeles R_id="123456789103">
        <kedvezmeny>0</kedvezmeny>
    </rendeles>
    <rendeles R_id="123456789104">
        <kedvezmeny>1500</kedvezmeny>
    </rendeles>

    <!-- rendelések vege -->
    <!-- vásárlók -->

    <vasarlo R_id_v="123456789101" V_id="1">
        <nev>Hajdu Tibor</nev>
        <telefonszam>06701234567</telefonszam>
        <cim>
            <irszam>4000</irszam>
            <telepules>Debrecen</telepules>
            <utca>Dobo Istvan utca</utca>
            <hazszam>45</hazszam>
        </cim>
    </vasarlo>
    <vasarlo R_id_v="123456789102" V_id="2">
        <nev>Kiss Lajos</nev>
        <telefonszam>06301234567</telefonszam>
        <cim>
            <irszam>3600</irszam>
            <telepules>Ozd</telepules>
            <utca>Lakat utca</utca>
            <hazszam>10</hazszam>
        </cim>
    </vasarlo>
    <vasarlo R_id_v="123456789103" V_id="3">
        <nev>Bodnár Peter</nev>
        <telefonszam>06201234567</telefonszam>
        <cim>
            <irszam>2000</irszam>
            <telepules>Szeged</telepules>
            <utca>Nadas utca</utca>
            <hazszam>2</hazszam>
        </cim>
    </vasarlo>

```

```

<vasarlo R_id_v="123456789104" V_id="4">
  <nev>Nagy Lujza</nev>
  <telefonszam>06301234666</telefonszam>
  <cim>
    <irszam>3500</irszam>
    <telepules>Miskolc</telepules>
    <utca>Leleenyves utca</utca>
    <hazszam>38</hazszam>
  </cim>
</vasarlo>

<!-- vasarlok vege -->
<!-- futarszolgalatok -->

<futarszolgalat R_id_fsz="123456789101" Fsz_id="11">
  <nev>Ups Futarszolgalat</nev>
</futarszolgalat>
<futarszolgalat R_id_fsz="123456789102" Fsz_id="12">
  <nev>MPL Futarszolgalat</nev>
</futarszolgalat>
<futarszolgalat R_id_fsz="123456789103" Fsz_id="13">
  <nev>ExpressOne Futarszolgalat</nev>
</futarszolgalat>
<futarszolgalat R_id_fsz="123456789104" Fsz_id="14">
  <nev>FOXPOST Futarszolgalat</nev>
</futarszolgalat>
<!-- futarszolgalatok vege -->
<!-- szamlak -->

<szamla R_id_sz="123456789101" SZ_id="001">
  <osszeg>50000</osszeg>
  <mennyiseg>2</mennyiseg>
  <datum>2021.10.02.</datum>
</szamla>
<szamla R_id_sz="123456789102" SZ_id="002">
  <osszeg>19500</osszeg>
  <mennyiseg>1</mennyiseg>
  <datum>2021.09.20.</datum>
</szamla>
<szamla R_id_sz="123456789103" SZ_id="0033">
  <osszeg>32000</osszeg>
  <mennyiseg>1</mennyiseg>
  <datum>2021.11.11.</datum>
</szamla>
<szamla R_id_sz="123456789104" SZ_id="004">
  <osszeg>76500</osszeg>
  <mennyiseg>3</mennyiseg>
  <datum>2021.08.23.</datum>
</szamla>

<!-- szamlak vege -->
</cipo_rendeles>

```

d) Az XML dokumentum alapján XMLSchema készítése

```

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

    <!-- Az XML dokumentum felepítése -->

    <xs:complexType>
      <xs:sequence>

```

```

        <xs:element name="cipo" type="cipoTipus" maxOccurs="unbounded" />
        <xs:element name="rendeles" type="rendelesTipus" maxOccurs="unbounded" />
        <xs:element name="vasarlo" type="vasarloTipus" maxOccurs="unbounded" />
        <xs:element name="fsz" type="fszTipus" maxOccurs="unbounded" />
        <xs:element name="szamla" type="szamlaTipus" maxOccurs="unbounded" />
    </xs:sequence>
</xs:complexType>

<!-- Kulcsok -->

<xs:key name="cipo_kulcs">
    <xs:selector xpath="cipo" />
    <xs:field xpath="@C_id"></xs:field>
</xs:key>
<xs:key name="rendeles_kulcs">
    <xs:selector xpath="rendeles" />
    <xs:field xpath="@R_id"></xs:field>
</xs:key>
<xs:key name="vasarlo_kulcs">
    <xs:selector xpath="vasarlo" />
    <xs:field xpath="@V_id"></xs:field>
</xs:key>
<xs:key name="fsz_kulcs">
    <xs:selector xpath="fsz" />
    <xs:field xpath="@Fsz_id"></xs:field>
</xs:key>
<xs:key name="szamla_kulcs">
    <xs:selector xpath="szamla"></xs:selector>
    <xs:field xpath="@SZ_id"></xs:field>
</xs:key>

<!-- idegen kulcsok -->

<xs:keyref refer="rendeles_kulcs" name="cipo_rendeles_idegen_kulcs">
    <xs:selector xpath="cipo"></xs:selector>
    <xs:field xpath="@R_id_c"></xs:field>
</xs:keyref>
<xs:keyref refer="rendeles_kulcs" name="vasarlo_rendeles_idegen_kulcs">
    <xs:selector xpath="vasarlo"></xs:selector>
    <xs:field xpath="@R_id_v"></xs:field>
</xs:keyref>
<xs:keyref refer="rendeles_kulcs" name="fsz_rendeles_idegen_kulcs">
    <xs:selector xpath="fsz"></xs:selector>
    <xs:field xpath="@R_id_fsz"></xs:field>
</xs:keyref>
<xs:keyref refer="rendeles_kulcs" name="szamla_rendeles_idegen_kulcs">
    <xs:selector xpath="szamla"></xs:selector>
    <xs:field xpath="@R_id_sz"></xs:field>
</xs:keyref>

<xs:unique name="unique_szamla">
    <xs:selector xpath="szamla"></xs:selector>
    <xs:field xpath="@R_id_sz"></xs:field>
</xs:unique>
</xs:element>

<!-- Tipusok -->
<!-- cipo tipus -->

<xs:complexType name="cipoTipus">
    <xs:sequence>
        <xs:element type="xs:string" name="marka"/>
        <xs:element type="xs:string" name="cipo_neve"/>
        <xs:element type="xs:short" name="ar"/>
        <xs:element type="xs:string" name="szin"/>
        <xs:element type="xs:string" name="keszlet"/>
    </xs:sequence>

```

```

        <xs:attribute type="xs:integer" name="cipo_id" use="required"/>
        <xs:attribute type="xs:integer" name="R_id_c" use="required"/>
    </xs:complexType>

    <!-- rendeles tipus -->

    <xs:complexType name="rendelesTipus">
        <xs:sequence>
            <xs:element type="xs:string" name="kedvezmeny"/>
        </xs:sequence>
        <xs:attribute type="xs:integer" name="R_id" 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="cim">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element type="xs:string" name="irszam"/>
                        <xs:element type="xs:string" name="telepules"/>
                        <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="V_id" use="required"/>
        <xs:attribute type="xs:integer" name="R_id_v" use="required"/>
    </xs:complexType>

    <!-- fsz tipus -->

    <xs:complexType name="fszTipus">
        <xs:sequence>
            <xs:element type="xs:string" name="nev"/>
        </xs:sequence>
        <xs:attribute type="xs:integer" name="fsz_id" use="required"/>
        <xs:attribute type="xs:integer" name="R_id_fsz" use="required"/>
    </xs:complexType>

    <!-- szamla tipus -->

    <xs:complexType name="szamlaTipus">
        <xs:sequence>
            <xs:element type="xs:string" name="mennyiseg"/>
            <xs:element type="xs:string" name="osszeg"/>
            <xs:element name="datum">
                <xs:simpleType>
                    <xs:restriction base="xs:gYear">
                        <xs:minInclusive value="2000"/>
                        <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

a) Adatolvasás

A program kimenetként megadja az XML dokumentumból kiolvasott tartalmat. Futtatáskor a DOMModifySvoxgh.java osztály is lefut, ami megkeresi és növeli a cipők árát 3000-rel. Miután megtörtént a módosítás, a program újra kiírja az adatokat, ezzel jelezve, hogy változás történt.

```
1 package hu.domparse.svoxgh;
2
3 import java.io.File;
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21 public class DOMReadSvoxgh {
22
23     private static void printNode(Node r) {
24
25         // Node nevének kiírása
26         if (r.getNodeName() != "#text") {
27             System.out.println(r.getNodeName());
28         }
29
30         // Gyermekek node listába helyezése
31         NodeList children = r.getChildNodes();
32         for (int i = 0; i < children.getLength(); i++) {
33             Node child = children.item(i);
34             boolean isComplex = child.getTextContent().contains("\n");
35
36             // Attribute kiírása
37             if (child.hasAttributes()) {
38                 NamedNodeMap attributes = child.getAttributes();
39                 int numAttrs = attributes.getLength();
40                 for (int j = 0; j < numAttrs; j++) {
41                     Attr attr = (Attr) attributes.item(j);
42                     String attrName = attr.getNodeName();
43                     String attrValue = attr.getNodeValue();
44                     System.out.println(" " + attrName + " : " + attrValue);
45                 }
46             }
47
48             // Nev és tartalom kiírása
49             if (isComplex) {
50                 printNode(child);
51             } else {
52                 System.out.print(" " + child.getNodeName());
53                 System.out.println(": " + child.getTextContent());
54             }
55         }
56     }
57 }
```



```

58 public static void main(String[] args) {
59     try {
60
61         DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
62         DocumentBuilder dbBuilder = dbFactory.newDocumentBuilder();
63         Document doc = dbBuilder.parse("src/hu/domparse/svoxgh/XMLSvoxgh.xml");
64         String filepath = "src/hu/domparse/svoxgh/XMLSvoxgh.xml";
65         doc.getDocumentElement().normalize();
66
67         // XPath
68         XPath xPath = XPathFactory.newInstance().newXPath();
69
70         Element root = doc.getDocumentElement();
71
72         System.out.println("\n ----- \n Modositas elott: \n ----- \n");
73
74         printNode(root);
75         DOMModifySvoxgh.modifyPrices(root);
76
77         System.out.println("\n ----- \n Modositas utan: \n ----- \n");
78
79         printNode(root);
80
81         // Modositott XML mentese
82         TransformerFactory transformerFactory = TransformerFactory.newInstance();
83         Transformer transformer = transformerFactory.newTransformer();
84         DOMSource source = new DOMSource(doc);
85         StreamResult result = new StreamResult(new File(filepath));
86         transformer.transform(source, result);
87
88     } catch (Exception e) {
89         e.printStackTrace();
90     }
91 }
92
93 }

```

b) Adatmódosítás

A program kikeresi a cipők árát és megnöveli azokat 3000-rel.

```

package hu.domparse.svoxgh;

import org.w3c.dom.Node;
import org.w3c.dom.NodeList;

public class DOMModifySvoxgh {

    public static void modifyPrices(Node root) {
        try {

            System.out.println(root.getNodeType());
            NodeList els = root.getChildNodes();

            // Vegigmegy a root (cipo_rendeles) gyerekein
            for (int j = 0; j < els.getLength(); j++) {

                // Ha a gyerek neve cipo, akkor vegrehajtja a kovetkezoiket
                if (els.item(j).getNodeName() == "cipo") {

                    // Vegigmegy a gyerek elemein
                    for (int k = 0; k <
                        els.item(j).getChildNodes().getLength(); k++) {

                        // Ha az egyik elem neve ar, akkor megnoveli annak erteket 3000-rel
                        if
                        (els.item(j).getChildNodes().item(k).getNodeName() == "ar") {

```

```

                                int ar =
Integer.parseInt(els.item(j).getChildNodes().item(k).getTextContent());

    els.item(j).getChildNodes().item(k).setTextContent(String.valueOf(ar + 3000));

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

```

c) Adatlekérdezés

1. lekérdezés: a program kilistázza azokat a márkákat, amelyek árulnak fekete cipőt. Ha nem található ilyen márka, a program jelzi azt.
2. lekérdezés: a program kilistázza azoknak a rendeléseknek az azonosítóját, amelyeket a GLS Futárszolgálat kézbesít. Ha nem található ilyen rendelés, a program jelzi azt.

```

1 package hu.dompars.svoxgh;
2
3 import java.io.File;
15
16 public class DOMQuerySvoxgh {
17
18     public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException {
19
20         File xmlFile = new File("src/hu/dompars/svoxgh/XMLSvoxgh.xml");
21
22         DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
23         DocumentBuilder dBuilder = factory.newDocumentBuilder();
24
25         Document doc = dBuilder.parse(xmlFile);
26
27         doc.getDocumentElement().normalize();
28
29         System.out.println("Root: " + doc.getDocumentElement().getNodeName() + "\n");
30
31         // Kiírja azokat a markákat, amelyekből van fekete színű cipő
32         System.out.println("A fekete színű cipők markái: \n");
33
34         NodeList nodeList = doc.getElementsByTagName("cipő");
35
36         boolean foundAny = false;
37
38         for(int i = 0; i < nodeList.getLength(); i++) {
39
40             Node node = nodeList.item(i);
41
42             if(node.getNodeType() == Node.ELEMENT_NODE) {
43                 Element elem = (Element) node;
44
45                 Node node2;
46
47                 // Megvizsgálom a cipő színt
48                 node2 = elem.getElementsByTagName("szín").item(0);
49                 String color = node2.getTextContent();
50
51                 // Ha fekete, akkor kiíratom a markát
52                 if("Fekete".equals(color)) {
53                     Node nodeName = elem.getElementsByTagName("marka").item(0);
54                     String brand = nodeName.getTextContent();
55
56                     System.out.println(brand + "\n");
57
58                     foundAny = true;
59                 }
60             }
61         }
62
63         if(foundAny == false) {
64
65             System.out.println("\n Nem található olyan marka, amely arul fekete cipőt. \n");
66         }
67         else {
68             foundAny = false;
69         }
70
71         // Kilistázza azoknak a rendeléseknek az azonosítóit, amelyeket a GLS Futárszolgálat kézbesít
72         System.out.println("\n GLS által kézbesítendő rendelések azonosítói: \n");
73
74         nodeList = doc.getElementsByTagName("futárszolgálat");
75
76         foundAny = false;
77
78         for(int i = 0; i < nodeList.getLength(); i++) {
79             Node node = nodeList.item(i);
80
81             if(node.getNodeType() == Node.ELEMENT_NODE) {
82                 Element elem = (Element) node;
83
84                 Node node2;
85
86                 // Megvizsgálja a futárszolgálatokat
87                 node2 = elem.getElementsByTagName("nev").item(0);
88                 String deliveryCompanyName = node2.getTextContent();
89
90                 // Ha GLS Futárszolgálat, akkor kiírja a rendelés azonosítóit
91                 if("GLS Futárszolgálat".equals(deliveryCompanyName)) {
92                     String order_id = elem.getAttribute("R_id_fsz");
93
94                     System.out.println(order_id + "\n");
95
96                     foundAny = true;
97                 }
98             }
99         }
100
101         if(foundAny == false) {
102             System.out.println("\n Nem található olyan rendelés az adatbázisban, amelyet a GLS Futárszolgálat kézbesít.\n");
103         }
104     }
105 }
106
107
108
109
110 }

```