

JEGYZŐKÖNYV

Adatkezelés XML környezetében

Féléves feladat

Könyvesbolt adatbázis

Készítette: **Tisza Marcell**

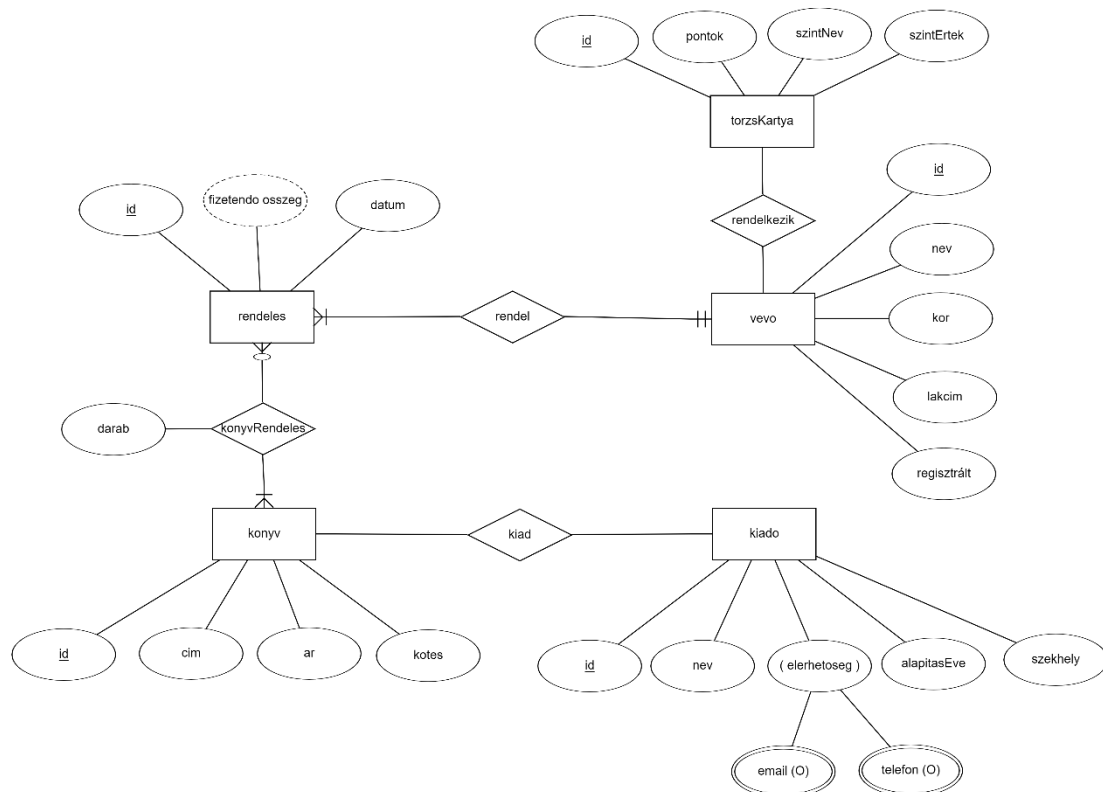
Neptunkód: **NLVHSR**

A feladat leírása:

Egy modern könyvesbolt adatbázisának a mintáját készítettem el, amely a bolt weboldalának a tartalmát képezi. Az adatbázis nyilvántartja a könyvesbolt által forgalmazott könyveket (információkat szolgáltat az árakról, valamint a könyvek kötésének típusáról stb.), a rendeléseket és a hozzájuk kapcsolódó vásárlói adatokat. Emellett a könyvek kiadóiról is tartalmaz néhány információt (mint pl. székhelyük vagy az alapításuk éve), amit majd az oldalon meg lehet jeleníteni. Tartalmazza a vásárlók adatait is, mint nevüket, korukat, lakcímüket, ahova majd a rendelést leszállíthatják, illetve tartalmazza még a regisztráció dátumát. Egy rendelés az adatbázisban mindig egy vevőhöz köthető, viszont egy vevő nevén lehet több rendelés is. Továbbá egy rendelés könyv(ek)et tartalmaz (egyet vagy többet). Ahol természetesen egy könyv több rendelésnek is része lehet, feltéve, hogy van elég belőle készleten. Mindemellett számon van még tartva, hogy egy könyv hányszor szerepel egy adott rendelésben. Ezen paramétereivel az adatbázis lehetőséget biztosít a vállalatnak, hogy nagyobb rálátása legyen az áruház készleteire, illetve a tranzakciók számontartására.

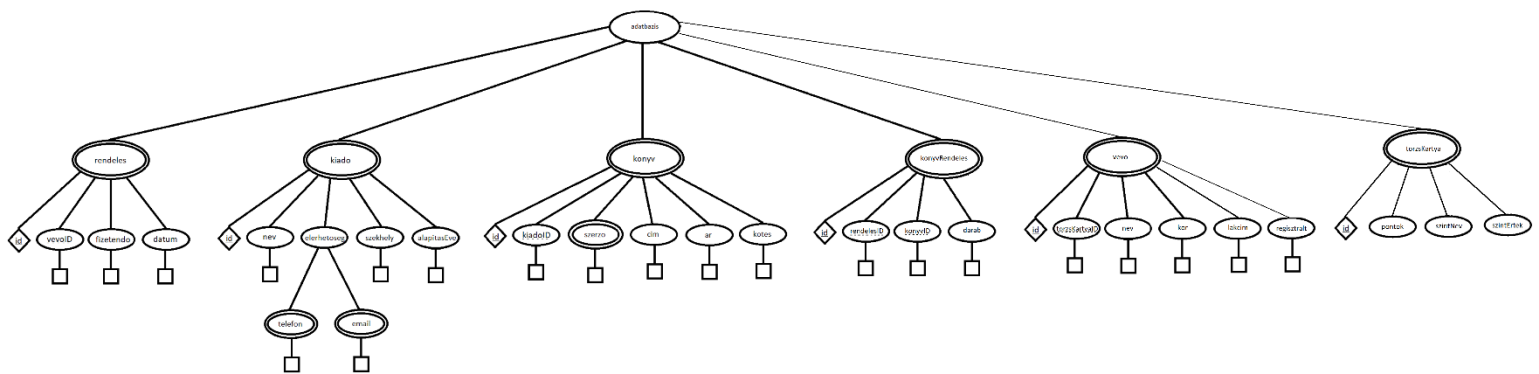
1. feladat

1a) Az adatbázis ER modell:



1b) Az adatbázis konvertálása XDM modellre

Az XDM modell az ER modelltől eltérően egy extra elemet is tartalmaz, névlegesen a könyvRendeles-t, amire azért van szükség, hogy meg tudjuk jeleníteni az ER modellben lévő N-M kapcsolatot. Továbbá ennek az elemnek lesz a tulajdonsága az a mennyiség, amennyit egy adott könyvből a kosárba rakott a vásárló. Az 1-N kapcsolatok a modellben kulcs-idegen kulcs párokként vannak megvalósítva. (ezt a kapcsolatot az xmlschema ellenőrzi). Ezeken a módosításokon kívül a modell többi tagja változatlan az ER modellhez képest



1c) Az XDM modell alapján XML dokumentum készítése

```
XMLNLVHSR.xml ×
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <adatbazis xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="XMLNLVHSR.xsd">
3
4 <!-- RENDELÉSEK -->
5 <rendeles id="r1">
6   <datum>2021-10-26</datum>
7   <fizetendo>2500</fizetendo>
8   <vevoID>v1</vevoID>
9
10 </rendeles>
11
12 <rendeles id="r2">
13   <datum>2021-10-30</datum>
14   <fizetendo>3200</fizetendo>
15   <vevoID>v2</vevoID>
16 </rendeles>
17
18 <rendeles id="r3">
19   <datum>2021-11-02</datum>
20   <fizetendo>1500</fizetendo>
21   <vevoID>v1</vevoID>
22 </rendeles>
23
24 <!-- KÖNYV RENDELÉSEK -->
25 <konyvRendeles id="kr1">
26   <rendelesID>r1</rendelesID>
27   <konyvID>k1</konyvID>
28   <darab>1</darab>
29 </konyvRendeles>
30
31 <konyvRendeles id="kr2">
32   <rendelesID>r1</rendelesID>
33   <konyvID>k2</konyvID>
34   <darab>2</darab>
35 </konyvRendeles>
36
37 <konyvRendeles id="kr3">
38   <rendelesID>r2</rendelesID>
39   <konyvID>k1</konyvID>
40   <darab>1</darab>
41 </konyvRendeles>
42
43 <konyvRendeles id="kr4">
44   <rendelesID>r2</rendelesID>
45   <konyvID>k2</konyvID>
46   <darab>1</darab>
47 </konyvRendeles>
48
49 <konyvRendeles id="kr5">
50   <rendelesID>r2</rendelesID>
51   <konyvID>k3</konyvID>
52   <darab>1</darab>
53 </konyvRendeles>
54
55 <konyvRendeles id="kr6">
56   <rendelesID>r3</rendelesID>
57   <konyvID>k3</konyvID>
58   <darab>1</darab>
59 </konyvRendeles>
60
61 <!-- VEVŐK -->
62 <vevo id="v1">
63   <nev>Olajos Lajos</nev>
64   <lakcim>1077 Budapest, Wesselényi u. 55</lakcim>
65   <kor>21</kor>
66   <torzsKartyaID>tk1</torzsKartyaID>
67   <regisztralt>2000-01-01</regisztralt>
68 </vevo>
69
70 <vevo id="v2">
71   <nev>Varga Katalin</nev>
72   <lakcim>3530 Miskolc, Rácz György u. 8</lakcim>
73   <kor>28</kor>
74   <torzsKartyaID>tk2</torzsKartyaID>
75   <regisztralt>2012-10-12</regisztralt>
76 </vevo>
```

```

77
78 <!-- TÖRZSKÁRTYÁK -->
79Ⓜ <torzsKartya id="tk1">
80   <pontok>10</pontok>
81   <szintNev>Normal</szintNev>
82   <szintErtek>1</szintErtek>
83 </torzsKartya>
84
85Ⓜ <torzsKartya id="tk2">
86   <pontok>3</pontok>
87   <szintNev>Szuper</szintNev>
88   <szintErtek>3</szintErtek>
89 </torzsKartya>
90
91 <!-- KÖNYVEK -->
92Ⓜ <konyv id="k1">
93   <szerzo>Szilasi László</szerzo>
94   <cim>Tavaszi hadjárat</cim>
95   <ar>5000</ar>
96   <kotes>kemény</kotes>
97   <kiadoID>kia1</kiadoID>
98 </konyv>
99
100Ⓜ <konyv id="k2">
101   <szerzo>Györffy Ákos</szerzo>
102   <cim>A távollodásban</cim>
103   <ar>500</ar>
104   <kotes>puha</kotes>
105   <kiadoID>kia1</kiadoID>
106 </konyv>
107
108Ⓜ <konyv id="k3">
109   <szerzo>Andrew S. Tanenbaum</szerzo>
110   <szerzo>David J. Wetherall</szerzo>
111   <cim>Számítógép-hálózatok</cim>
112   <ar>5000</ar>
113   <kotes>kemény</kotes>
114   <kiadoID>kia1</kiadoID>
115 </konyv>
116
117 <!-- KIADÓK -->
118Ⓜ <kiado id="kia1">
119   <nev>Magvető Könyvkiadó</nev>
120   <szekhely>Budapest</szekhely>
121   <alapitasEve>1955</alapitasEve>
122Ⓜ <elerhetoseg>
123   <telefon>+36-1/323-3981</telefon>
124   <telefon>+36-1/235-5032</telefon>
125   <email>magveto.kiado@lira.hu</email>
126 </elerhetoseg>
127 </kiado>
128
129Ⓜ <kiado id="kia2">
130   <nev>Panem</nev>
131   <szekhely>Budapest</szekhely>
132   <alapitasEve>1972</alapitasEve>
133Ⓜ <elerhetoseg>
134   <telefon>+36-1/460-0272</telefon>
135   <email>webbolt@panem.hu</email>
136 </elerhetoseg>
137 </kiado>
138
139 </adatbazis>

```

1d) Az XML dokumentum alapján XMLSchema készítése

```
XMLNLVHSR.xsd x
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="adatbazis">
4     <xs:complexType>
5       <xs:sequence>
6         <xs:element name="rendeles" type="rendelesTipus" minOccurs="0" maxOccurs="unbounded"/>
7         <xs:element name="konyvRendeles" type="konyvRendelesTipus" maxOccurs="unbounded"/>
8         <xs:element name="vevo" type="vevoTipus" maxOccurs="unbounded"/>
9         <xs:element name="torzsKartya" type="torzsKartyaTipus" maxOccurs="unbounded"/>
10        <xs:element name="konyv" type="konyvTipus" maxOccurs="unbounded"/>
11        <xs:element name="kiado" type="kiadoTipus" maxOccurs="unbounded"/>
12      </xs:sequence>
13    </xs:complexType>
14    <xs:key name="vevoKulcs">
15      <xs:selector xpath="vevo"/>
16      <xs:field xpath="@id"/>
17    </xs:key>
18    <xs:key name="torzsKartyaKulcs">
19      <xs:selector xpath="torzsKartya"/>
20      <xs:field xpath="@id"/>
21    </xs:key>
22    <xs:key name="konyvKulcs">
23      <xs:selector xpath="konyv"/>
24      <xs:field xpath="@id"/>
25    </xs:key>
26    <xs:key name="kiadoKulcs">
27      <xs:selector xpath="kiado"/>
28      <xs:field xpath="@id"/>
29    </xs:key>
30    <xs:key name="rendelesKulcs">
31      <xs:selector xpath="rendeles"/>
32      <xs:field xpath="@id"/>
33    </xs:key>
34    <xs:keyref name="vevo-torzsKartya" refer="torzsKartyaKulcs">
35      <xs:selector xpath="vevo/torzsKartyaID"/>
36      <xs:field xpath="."/>
37    </xs:keyref>
38    <xs:keyref name="rendeles-vevo" refer="vevoKulcs">
39      <xs:selector xpath="rendeles/vevoID"/>
40      <xs:field xpath="."/>
41    </xs:keyref>
42    <xs:keyref name="konyvRendeles-konyv" refer="konyvKulcs">
43      <xs:selector xpath="konyvRendeles/konyvID"/>
44      <xs:field xpath="."/>
45    </xs:keyref>
46    <xs:keyref name="konyv-kiado" refer="kiadoKulcs">
47      <xs:selector xpath="konyv/kiadoID"/>
48      <xs:field xpath="."/>
49    </xs:keyref>
50    <xs:keyref name="konyvRendeles-rendeles" refer="rendelesKulcs">
51      <xs:selector xpath="konyvRendeles/rendelesID"/>
52      <xs:field xpath="."/>
53    </xs:keyref>
54  </xs:element>
55
56  <xs:complexType name="vevoTipus">
57    <xs:sequence>
58      <xs:element name="nev" type="xs:string"/>
59      <xs:element name="lakcim" type="xs:string"/>
60      <xs:element name="kor" type="xs:unsignedInt"/>
61      <xs:element name="torzsKartyaID" type="xs:string"/>
62      <xs:element name="regisztralt" type="xs:date"/>
63    </xs:sequence>
64    <xs:attribute name="id" type="xs:ID" use="required"/>
65  </xs:complexType>
66
67  <xs:complexType name="torzsKartyaTipus">
68    <xs:sequence>
69      <xs:element name="pontok" type="xs:string"/>
70      <xs:element name="szintNev" type="xs:string"/>
71      <xs:element name="szintErtek" type="xs:unsignedInt"/>
72    </xs:sequence>
73    <xs:attribute name="id" type="xs:ID" use="required"/>
74  </xs:complexType>
75
76  <xs:complexType name="konyvTipus">
77    <xs:sequence>
78      <xs:element name="szerzo" type="xs:string" maxOccurs="unbounded"/>
79      <xs:element name="cim" type="xs:string"/>
80      <xs:element name="ar" type="xs:int"/>
81      <xs:element name="kotes" type="kotesTipus"/>
82      <xs:element name="kiadoID" type="xs:string"/>
83    </xs:sequence>
```

```

84     <xs:attribute name="id" type="xs:ID" use="required"/>
85 </xs:complexType>
86
87 <xs:simpleType name="kotesTipus">
88     <xs:restriction base="xs:string">
89         <xs:enumeration value="kemeny"/>
90         <xs:enumeration value="puha"/>
91     </xs:restriction>
92 </xs:simpleType>
93
94 <xs:complexType name="kiadoTipus">
95     <xs:sequence>
96         <xs:element name="nev" type="xs:string"/>
97         <xs:element name="szekhely" type="xs:string"/>
98         <xs:element name="alapitasEve" type="xs:unsignedInt"/>
99         <xs:element name="elerhetoseg" type="elerhetosegTipus"/>
100     </xs:sequence>
101     <xs:attribute name="id" type="xs:ID" use="required"/>
102 </xs:complexType>
103
104 <xs:complexType name="elerhetosegTipus">
105     <xs:sequence>
106         <xs:element name="telefon" type="telefonTipus" minOccurs="1" maxOccurs="unbounded"/>
107         <xs:element name="email" type="emailTipus" minOccurs="0" maxOccurs="unbounded"/>
108     </xs:sequence>
109 </xs:complexType>
110
111 <xs:simpleType name="emailTipus">
112     <xs:restriction base="xs:string">
113         <xs:pattern value="[ \w. ]+@ \w+ \. \w+ "/>
114     </xs:restriction>
115 </xs:simpleType>
116
117 <xs:simpleType name="telefonTipus">
118     <xs:restriction base="xs:string">
119         <xs:pattern value="\+36- \d{1,2}/ \d{3}- \d{4}"/>
120     </xs:restriction>
121 </xs:simpleType>
122
123 <xs:complexType name="rendelesTipus">
124     <xs:sequence>
125         <xs:element name="datum" type="xs:date"/>
126         <xs:element name="fizetendo" type="xs:unsignedInt"/>
127         <xs:element name="vevoID" type="xs:string"/>
128     </xs:sequence>
129     <xs:attribute name="id" type="xs:ID" use="required"/>
130 </xs:complexType>
131
132 <xs:complexType name="konyvRendelesTipus">
133     <xs:sequence>
134         <xs:element name="rendelesID" type="xs:string"/>
135         <xs:element name="konyvID" type="xs:string"/>
136         <xs:element name="darab" type="xs:unsignedInt"/>
137     </xs:sequence>
138     <xs:attribute name="id" type="xs:ID" use="required"/>
139 </xs:complexType>
140 </xs:schema>

```

2. feladat

2a) adatolvasás – DOMReadNLVHSR.java

A könyvesbolt adatbázisának a formázott kiírása. A program típusonként csoportosítva írja ki az egyes entitásokat, kiemelve az azonosítójukat a könnyebb kezelhetőség érdekében.

```
DOMReadNLVHSR.java x
1 package hu.domp.parse.nlvhsr;
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.Node;
13 import org.w3c.dom.NodeList;
14 import org.xml.sax.SAXException;
15
16 public class DOMReadNLVHSR {
17
18     public static void main(String[] args) {
19         try {
20
21             // A DOM objektum létrehozása az XML dokumentumból
22             File xmlFile = new File("XMLNLVHSR.xml");
23             DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
24             DocumentBuilder dBuilder = factory.newDocumentBuilder();
25             Document doc = dBuilder.parse(xmlFile);
26             doc.getDocumentElement().normalize();
27
28             // Gyökér elem kiírása (adatbázis)
29             System.out.println("Root element: " + doc.getDocumentElement().getNodeName() + "\n");
30
31             // Az adatbázis elemeinek kiírása tulajdonságaikkal együtt formázva
32             String[] tagNames = { "vevo", "konyv", "kiado", "rendeles", "konyvRendeles", "torzsKartya" };
33             for (String tagName : tagNames) {
34                 NodeList nodeList = doc.getElementsByTagName(tagName);
35                 String message = tagName + " elemek:";
36                 System.out.println("\n" + message);
37                 System.out.println("-".repeat(message.length()) + "\n");
38
39                 // Elemek tulajdonságainak és azonosítójának kiírása
40                 for (int i = 0; i < nodeList.getLength(); i++) {
41                     Node nNode = nodeList.item(i);
42                     if (nNode.getNodeType() == Node.ELEMENT_NODE) {
43                         Element elem = (Element)nNode;
44
45                         // Azonosító kiírása
46                         String id = elem.getAttribute("id");
47                         System.out.println(">>> ID: " + id);
48
49                         // Tulajdonságok (gyermek elemek) kiírása
50                         String nodeContent = "";
51                         NodeList childNodes = elem.getChildNodes();
52                         for (int j = 0; j < childNodes.getLength(); j++) {
53                             if (childNodes.item(j).getTextContent().trim() != "") {
54                                 nodeContent = normalizeText(childNodes.item(j).getTextContent().trim());
55                                 System.out.println(childNodes.item(j).getNodeName() + ": " + nodeContent);
56                             }
57                         }
58                     }
59                 }
60                 System.out.println();
61             }
62
63             // Esetleges hibák kezelése
64         } catch (SAXException | IOException | ParserConfigurationException ex) {
65             System.out.println("Some error occurred\nDescription:\n" + ex.getMessage());
66             ex.printStackTrace();
67         }
68     }
69
70     // Szöveg formázása a szép megjelenés érdekében
71     private static String normalizeText(String text) {
72         text = text.replaceAll("\n", " ");
73         text = text.replaceAll("\s+", " ");
74         return text;
75     }
76
77 }
```


Output:

```
Root element: adatbazis

vevo elemek:
-----

>>> ID: v1
nev: Olajos Lajos
lakcim: 1077 Budapest, Wesselényi u. 55
kor: 21
torzsKartyaID: tk1
regisztralt: 2000-01-01

>>> ID: v2
nev: Varga Katalin
lakcim: 3530 Miskolc, Rác György u. 8
kor: 28
torzsKartyaID: tk2
regisztralt: 2012-10-12

konyv elemek:
-----

>>> ID: k1
szerzo: Szilasi László
cim: Tavaszi hadjárat
ar: 5000
kotes: kemeny
kiadoID: kia1

>>> ID: k2
szerzo: Györfly Ákos
cim: A távolodásban
ar: 500
kotes: puha
kiadoID: kia1

>>> ID: k3
szerzo: Andrew S. Tanenbaum
szerzo: David J. Wetherall
cim: Számítógép-hálózatok
ar: 5000
kotes: kemeny
kiadoID: kia1

kiado elemek:
-----

>>> ID: kia1
nev: Magvető Könyvkiadó
szekhely: Budapest
alapitasEve: 1955
elerhetoseg: +36-1/323-3981, +36-1/235-5032, magveto.kiado@lira.hu

>>> ID: kia2
nev: Panem
szekhely: Budapest
alapitasEve: 1972
elerhetoseg: +36-1/460-0272, webbolt@panem.hu

rendeles elemek:
-----

>>> ID: r1
datum: 2021-10-26
fizetendo: 2500
vevoID: v1

>>> ID: r2
datum: 2021-10-30
fizetendo: 3200
```

```
vevoID: v2

>>> ID: r3
datum: 2021-11-02
fizetendo: 1500
vevoID: v1

konyvRendeles elemek:
-----

>>> ID: kr1
rendelesID: r1
konyvID: k1
darab: 1

>>> ID: kr2
rendelesID: r1
konyvID: k2
darab: 2

>>> ID: kr3
rendelesID: r2
konyvID: k1
darab: 1

>>> ID: kr4
rendelesID: r2
konyvID: k2
darab: 1

>>> ID: kr5
rendelesID: r2
konyvID: k3
darab: 1

>>> ID: kr6
rendelesID: r3
konyvID: k3
darab: 1

torzsKartya elemek:
-----

>>> ID: tk1
pontok: 10
szintNev: Normal
szintErtek: 1

>>> ID: tk2
pontok: 3
szintNev: Szuper
szintErtek: 3
```

2b) adatmódosítás – DOMModifyNLVHSR.java

A program 3 módosítást végez:

- Módosítja a v1-es azonosítójú vásárló életkorát 19-re
- Módosítja a k1-es azonosítójú könyv árát 5000 Ft-ra
- Kibővíti a kia1-es azonosítójú kiadó elérhetőségeit a valami@gmail.com email címmel

```
DOMModifyNLVHSR.java x
1 package hu.domparse.nlvhsr;
2
3 import java.io.File;
4 import java.io.IOException;
5 import java.io.UnsupportedEncodingException;
6
7 import javax.xml.parsers.DocumentBuilder;
8 import javax.xml.parsers.DocumentBuilderFactory;
9 import javax.xml.parsers.ParserConfigurationException;
10 import javax.xml.transform.OutputKeys;
11 import javax.xml.transform.Transformer;
12 import javax.xml.transform.TransformerException;
13 import javax.xml.transform.TransformerFactory;
14 import javax.xml.transform.dom.DOMSource;
15 import javax.xml.transform.stream.StreamResult;
16
17 import org.w3c.dom.Document;
18 import org.w3c.dom.Element;
19 import org.w3c.dom.Node;
20 import org.w3c.dom.NodeList;
21 import org.xml.sax.SAXException;
22
23 public class DOMModifyNLVHSR {
24
25     public static void main(String[] args) {
26         try {
27             // A DOM objektum létrehozása az XML dokumentumból
28             File xmlFile = new File("XMLNLVHSR.xml");
29             DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
30             DocumentBuilder dBuilder = factory.newDocumentBuilder();
31             Document document = dBuilder.parse(xmlFile);
32             document.getDocumentElement().normalize();
33
34             // A v1-es azonosítójú vásárló korának módosítása 19-re
35             NodeList nodes = document.getElementsByTagName("vevo");
36             for (int i = 0; i < nodes.getLength(); i++) {
37                 Node node = nodes.item(i);
38                 if (node.getNodeType() == Node.ELEMENT_NODE) {
39                     if (node.getAttribute("id").getTextContent().equals("v1")) {
40                         NodeList childNodes = node.getChildNodes();
41                         for (int j = 0; j < childNodes.getLength(); j++) {
42                             Node childNode = childNodes.item(j);
43                             if (childNode.getNodeName().equals("kor")) {
44                                 childNode.setTextContent("19");
45                             }
46                         }
47                     }
48                 }
49             }
50
51             // Az k1-es azonosítójú könyv árának módosítása 5000-re
52             nodes = document.getElementsByTagName("konyv");
53             for (int i = 0; i < nodes.getLength(); i++) {
54                 Node node = nodes.item(i);
55                 if (node.getNodeType() == Node.ELEMENT_NODE) {
56                     if (node.getAttribute("id").getTextContent().equals("k1")) {
57                         NodeList childNodes = node.getChildNodes();
58                         for (int j = 0; j < childNodes.getLength(); j++) {
59                             Node childNode = childNodes.item(j);
60                             if (childNode.getNodeName().equals("ar")) {
61                                 childNode.setTextContent("5000");
62                             }
63                         }
64                     }
65                 }
66             }
67
68             // A kia1-es kiadó elérhetőségeinek a kibővítése a valami@gmail.com email címmel
69             nodes = document.getElementsByTagName("kiado");
70             for (int i = 0; i < nodes.getLength(); i++) {
71                 Node node = nodes.item(i);
72                 if (node.getNodeType() == Node.ELEMENT_NODE) {
73                     if (node.getAttribute("id").getTextContent().equals("kia1")) {
74                         NodeList childNodes = node.getChildNodes();
75                         for (int j = 0; j < childNodes.getLength(); j++) {
76                             Node childNode = childNodes.item(j);
77                             if (childNode.getNodeName().equals("elerhetoseg")) {
78                                 Element newElement = document.createElement("email");
79                                 newElement.setTextContent("valami@gmail.com");
80                                 childNode.appendChild(newElement);
81                             }
82                         }
83                     }
84                 }
85             }
86         }
87     }
88 }
```

```

86
87         // Kiíratás
88         File myFile = new File("XMLNLVHSR.xml");
89         writeXml(document, myFile);
90     }
91     // Esetleges hibák kezelése
92     catch (ParserConfigurationException | SAXException | IOException | TransformerException ex) {
93         System.out.println("Some error occurred\nDescription:\n" + ex.getMessage());
94         ex.printStackTrace();
95     }
96 }
97
98 // A módosított XML dokumentum kiíratása fájlba és a konzolra
99 private static void writeXml(Document doc, File output) throws TransformerException, UnsupportedEncodingException {
100     TransformerFactory transformerFactory = TransformerFactory.newInstance();
101     Transformer transf = transformerFactory.newTransformer();
102     transf.setOutputProperty(OutputKeys.ENCODING, "UTF-8");
103     transf.setOutputProperty(OutputKeys.INDENT, "yes");
104     transf.setOutputProperty("{http://xml.apache.org/xslt}indent-amunt", "2");
105
106     DOMSource source = new DOMSource(doc);
107
108     StreamResult console = new StreamResult(System.out);
109     StreamResult file = new StreamResult(output);
110
111     transf.transform(source, console);
112     transf.transform(source, file);
113 }
114 }

```

Output:

```

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

    <!-- RENDELÉSSEK -->

    <rendeles id="r1">

        <datum>2021-10-26</datum>

        <fizetendo>2500</fizetendo>

        <vevoID>v1</vevoID>

    </rendeles>

    <rendeles id="r2">

        <datum>2021-10-30</datum>

        <fizetendo>3200</fizetendo>

        <vevoID>v2</vevoID>

    </rendeles>

    <rendeles id="r3">

        <datum>2021-11-02</datum>

        <fizetendo>1500</fizetendo>

        <vevoID>v1</vevoID>

    </rendeles>

    <!-- KÁ-nyv RENDELÉSSEK -->

    <konyvRendeles id="kr1">

        <rendelesID>r1</rendelesID>

        <konyvID>k1</konyvID>

        <darab>1</darab>

    </konyvRendeles>

```

```
<konyvRendeles id="kr2">

    <rendelesID>r1</rendelesID>

    <konyvID>k2</konyvID>

    <darab>2</darab>

</konyvRendeles>

<konyvRendeles id="kr3">

    <rendelesID>r2</rendelesID>

    <konyvID>k1</konyvID>

    <darab>1</darab>

</konyvRendeles>

<konyvRendeles id="kr4">

    <rendelesID>r2</rendelesID>

    <konyvID>k2</konyvID>

    <darab>1</darab>

</konyvRendeles>

<konyvRendeles id="kr5">

    <rendelesID>r2</rendelesID>

    <konyvID>k3</konyvID>

    <darab>1</darab>

</konyvRendeles>

<konyvRendeles id="kr6">

    <rendelesID>r3</rendelesID>

    <konyvID>k3</konyvID>

    <darab>1</darab>
```

```

</konyvRendeles>

<!-- VEVLÉK -->

<vevo id="v1">

    <nev>Olajos Lajos</nev>

    <lakcim>1077 Budapest, Mészáros u. 55</lakcim>

    <kor>19</kor>

    <torzsKartyaID>tk1</torzsKartyaID>

    <regisztralt>2000-01-01</regisztralt>

</vevo>

<vevo id="v2">

    <nev>Varga Katalin</nev>

    <lakcim>3530 Miskolc, Rákóczi György u. 8</lakcim>

    <kor>28</kor>

    <torzsKartyaID>tk2</torzsKartyaID>

    <regisztralt>2012-10-12</regisztralt>

</vevo>

<!-- TÁ-RZSKÁRTYÁK -->
<torzsKartya id="tk1">
    <pontok>10</pontok>
    <szintNev>Normal</szintNev>
    <szintErtek>1</szintErtek>
</torzsKartya>

<torzsKartya id="tk2">
    <pontok>3</pontok>
    <szintNev>Szuper</szintNev>
    <szintErtek>3</szintErtek>
</torzsKartya>

<!-- KÁ-NYVEK -->
<konyv id="k1">
    <szerzo>Szilasi László</szerzo>
    <cim>Tavaszi hadjárat</cim>
    <ar>5000</ar>
    <kotes>kemény</kotes>
    <kiadoID>kia1</kiadoID>

</konyv>

```


2c) adatlekérdezés – DOMQueryNLVHSR.java

A program 3 lekérdezést hajt végre:

- 3000 Ft-nál nagyobb értékű rendelések adatai
- A v2-es azonosítójú vásárló adatai
- A 20 és 30 év közötti vásárlók adatai

```
DOMQueryNLVHSR.java ×
1 package hu.domparsenlvhsr;
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.xpath.XPath;
10 import javax.xml.xpath.XPathConstants;
11 import javax.xml.xpath.XPathExpressionException;
12 import javax.xml.xpath.XPathFactory;
13
14 import org.w3c.dom.Document;
15 import org.w3c.dom.Element;
16 import org.w3c.dom.Node;
17 import org.w3c.dom.NodeList;
18 import org.xml.sax.SAXException;
19
20 public class DOMQueryNLVHSR {
21
22     public static void main(String[] args) {
23         try {
24             // A DOM objektum létrehozása az XML dokumentumból
25             File xmlFile = new File("XMLNLVHSR.xml");
26             DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
27             DocumentBuilder dBuilder = factory.newDocumentBuilder();
28             Document document = dBuilder.parse(xmlFile);
29             document.getDocumentElement().normalize();
30
31             String message = "3000 Ft-nál nagyobb értékű rendelések:";
32             System.out.println(message + "\n" + "-".repeat(message.length()));
33             Query1(document);
34             message = "v2 azonosítójú vásárló:";
35             System.out.println(message + "\n" + "-".repeat(message.length()));
36             Query2(document);
37             message = "20 és 30 év közötti vásárlók:";
38             System.out.println(message + "\n" + "-".repeat(message.length()));
39             Query3(document);
40         }
41         // Esetleges hibák kezelése
42         catch(ParserConfigurationException | IOException | SAXException ex) {
43             System.out.println("Some error occurred\nDescription:\n" + ex.getMessage());
44             ex.printStackTrace();
45         }
46     }
47
48     // Szöveg formázása a szép megjelenés érdekében
49     private static String normalizeText(String text) {
50         text = text.replaceAll("\n", " ");
51         text = text.replaceAll("\\s+", " ");
52         return text;
53     }
54
55     // 3000 Ft-nál nagyobb értékű rendelések
56     private static void Query1(Document doc) {
57         NodeList rendelesek = doc.getElementsByTagName("rendeles");
58         for (int i = 0; i < rendelesek.getLength(); i++) {
59             Element rendeles = (Element)rendelesek.item(i);
60             NodeList childNodes = rendeles.getChildNodes();
61             for (int j = 0; j < childNodes.getLength(); j++) {
62                 Node childNode = childNodes.item(j);
63                 if (childNode.getNodeName().equals("fizetendo")) {
64                     if (Integer.parseInt(childNode.getTextContent()) > 3000)
65                         printElement(rendeles);
66                 }
67             }
68         }
69     }
70
71     // v2 azonosítójú vásárló
72     private static void Query2(Document doc) {
73         NodeList vevok = doc.getElementsByTagName("vevo");
74         for (int i = 0; i < vevok.getLength(); i++) {
75             Element vevo = (Element)vevok.item(i);
76             if (vevo.getAttributeNode("id").getValue().equals("v1"))
77                 printElement(vevo);
78         }
79     }
80
81     // 20 és 30 év közötti vásárlók
82     private static void Query3(Document doc) {
83         NodeList vevok = doc.getElementsByTagName("vevo");
84         for (int i = 0; i < vevok.getLength(); i++) {
85             Element vevo = (Element)vevok.item(i);
86             NodeList childNodes = vevo.getChildNodes();
87             for (int j = 0; j < childNodes.getLength(); j++) {
```



```

87         Node childNode = childNodes.item(j);
88         if (childNode.getNodeName().equals("kor") ) {
89             int kor = Integer.parseInt(childNode.getTextContent());
90             if (kor > 20 && kor < 30)
91                 printElement(vevo);
92         }
93     }
94 }
95 }
96
97 private static void printElement(Element elem) {
98     // Azonosító kiírása
99     String id = elem.getAttribute("id");
100    System.out.println(">>> ID: " + id);
101
102    // Tulajdonságok (gyermek elemek) kiírása
103    String nodeContent = "";
104    NodeList childNodes = elem.getChildNodes();
105    for (int j = 0; j < childNodes.getLength(); j++) {
106        if (childNodes.item(j).getTextContent().trim() != "") {
107            nodeContent = normalizeText(childNodes.item(j).getTextContent().trim());
108            System.out.println(childNodes.item(j).getNodeName() + ": " + nodeContent);
109        }
110    }
111    System.out.println();
112 }
113 }

```

Output:

```

3000 Ft-nál nagyobb értékű rendelések:
-----
>>> ID: r2
datum: 2021-10-30
fizetendo: 3200
vevoID: v2

v2 azonosítójú vásárló:
-----
>>> ID: v1
nev: Olajos Lajos
lakcim: 1077 Budapest, Wesselényi u. 55
kor: 21
torzsKartyaID: tk1
regisztralt: 2000-01-01

20 és 30 év közötti vásárlók:
-----
>>> ID: v1
nev: Olajos Lajos
lakcim: 1077 Budapest, Wesselényi u. 55
kor: 21
torzsKartyaID: tk1
regisztralt: 2000-01-01

>>> ID: v2
nev: Varga Katalin
lakcim: 3530 Miskolc, Rácz György u. 8
kor: 28
torzsKartyaID: tk2
regisztralt: 2012-10-12

```