

Ételrendelés és hozzá tartozó adatok-
Adatkezelés XML környezetben
Féléves feladat
Jegyzőkönyv

Készítette: Nádházy Gergely

Neptun-kód: ICDOQV

Feladat leírása

Féléves feladatomban egy ételrendelés menetét szeretném bemutatni, amely számon tart vevőket, rendeléseket, éttermeket, az ott dolgozó futárokat illetve a vevőkhöz tartozó bankkártya adatait.

Az adatbázis egyedei és tulajdonságai

Bankkártya egyedhez tartozik:

- kártyaszám
- tulajdonos neve
- illetve a lejárat dátum.
- kibocsátó bank neve és a kártya típusa, mindkét tulajdonság string típusú.

Vevőkhöz tartozik:

- a név, mely nyilvánvaló tulajdonság.
- telefonszám
- cím, amely egy összetett tulajdonság, tartalmazza az irányítószámot, várost, utcát és házszámot.

A rendeléshez tartozik:

- a teljes ár
- időpont, amikor megrendelték.

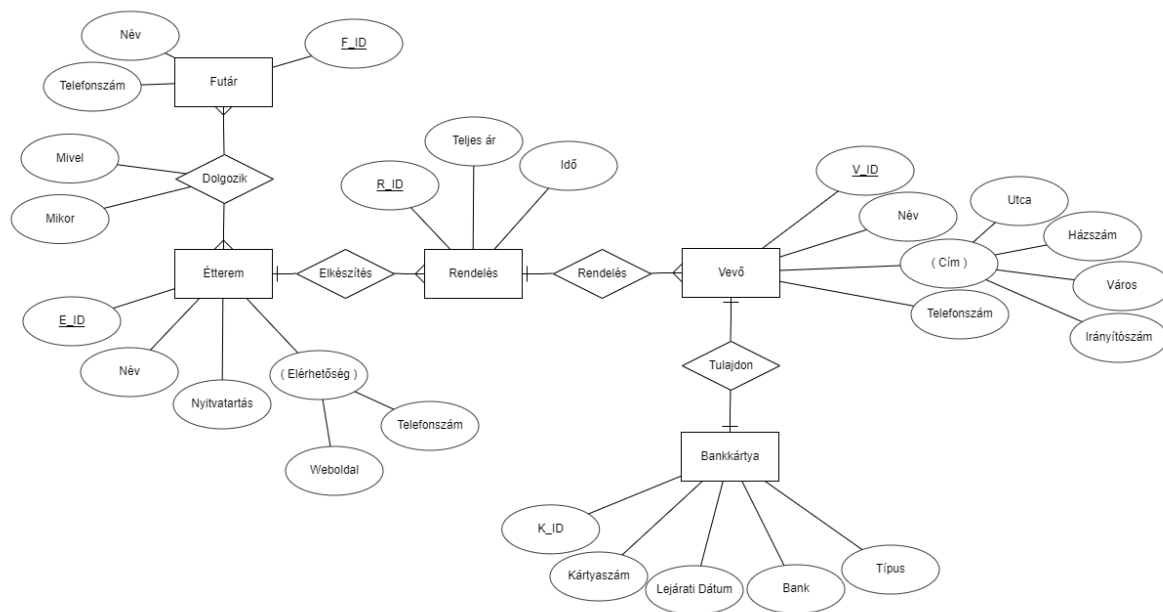
A futár tulajdonságai:

- telefonszáma egy integer
- neve egy string.

Az étterem mező tulajdonságai:

- Név, string
- Elérhetőség, mely tartalmazza a weboldalt és telefonszámot
- Nyitva tartás, ami string típusú

Az adatbázis ER modellje:

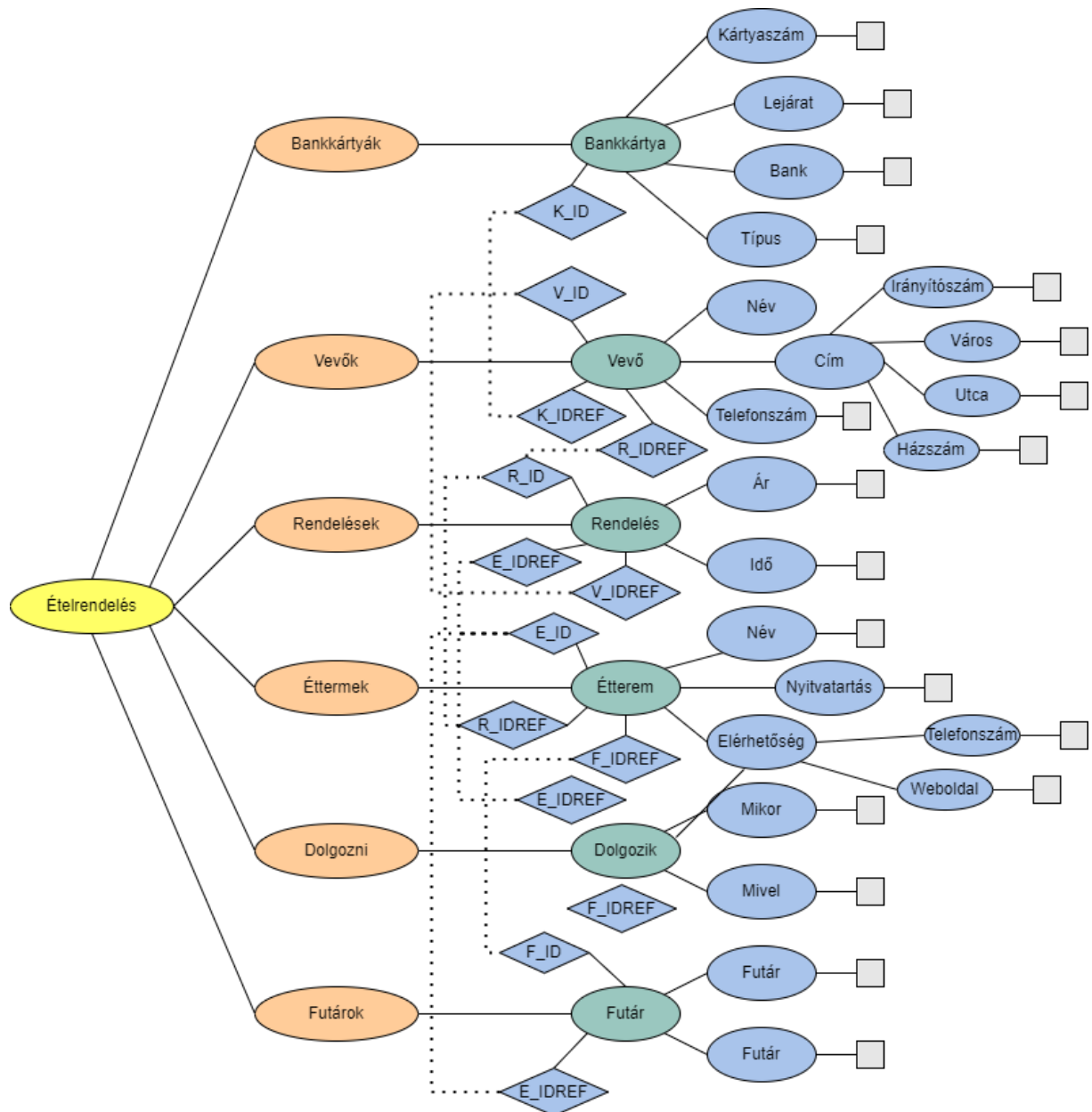


Konvertálás XDM-modellre:

Az ER modell konvertálása eredményezi az idegen kulcsok létrejöttét, ezek referálnak a hozzájuk tartozó elsődleges kulcsára.

Ebben az esetben a több a többhöz kapcsolatnál, amely a Dolgozik tábla lesz, ami összeköti a futárokat és az éttermet, egy új táblát vezetünk be.

A kész XDM modell:



XML file létrehozása az XDM modell alapján:

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <?xml-model href="XMLSchemaICDOQV.xsd" type="application/xml" schematypens="http://www.w3.org/2001/XMLSchema"?>
3
4  <etelrendeles>
5    <bankkartyak>
6      <bankkartya K_ID="1">
7        <kartyaszam>5321804176549876</kartyaszam>
8        <lejarat>08/23</lejarat>
9        <bank>OTP Bank</bank>
10       <tipus>Junior</tipus>
11     </bankkartya>
12     <bankkartya K_ID="2">
13       <kartyaszam>6543778237610094</kartyaszam>
14       <lejarat>11/22</lejarat>
15       <bank>Erste Bank</bank>
16       <tipus>Jump</tipus>
17     </bankkartya>
18     <bankkartya K_ID="3">
19       <kartyaszam>5321804198812560</kartyaszam>
20       <lejarat>04/23</lejarat>
21       <bank>OTP Bank</bank>
22       <tipus>Hitel</tipus>
23     </bankkartya>
24   </bankkartyak>
25
```

```
26   <vevok>
27     <vevo V_ID="1" K_IDREF="3" R_IDREF="5">
28       <nev>Sirankó Boldizsár</nev>
29       <cim>
30         <iranyitoszam>9432</iranyitoszam>
31         <varos>Kaposvár</varos>
32         <utca>Dózsa György utca</utca>
33         <hazszam>32</hazszam>
34       </cim>
35       <telefonszam>06303658765</telefonszam>
36     </vevo>
37     <vevo V_ID="2" K_IDREF="1" R_IDREF="3">
38       <nev>Timkó Attila</nev>
39       <cim>
40         <iranyitoszam>4500</iranyitoszam>
41         <varos>Ózd</varos>
42         <utca>Bánya út</utca>
43         <hazszam>45</hazszam>
44       </cim>
45       <telefonszam>06307629813</telefonszam>
46     </vevo>
47     <vevo V_ID="3" K_IDREF="2" R_IDREF="1">
48       <nev>Szabó Zoltán</nev>
49       <cim>
50         <iranyitoszam>7640</iranyitoszam>
51         <varos>Hajdúnánás</varos>
52         <utca>Mező utca</utca>
53         <hazszam>2</hazszam>
54       </cim>
55       <telefonszam>06709463371</telefonszam>
56     </vevo>
57   </vevok>
58
```

```

59     <rendeles>
60         <rendeles R_ID="1" V_IDREF="2" E_IDREF="1">
61             <ar>2690</ar>
62             <ido>11:32</ido>
63         </rendeles>
64         <rendeles R_ID="2" V_IDREF="1" E_IDREF="2">
65             <ar>9990</ar>
66             <ido>13:52</ido>
67         </rendeles>
68         <rendeles R_ID="3" V_IDREF="3" E_IDREF="1">
69             <ar>3450</ar>
70             <ido>14:14</ido>
71         </rendeles>
72         <rendeles R_ID="4" V_IDREF="2" E_IDREF="3">
73             <ar>10550</ar>
74             <ido>15:22</ido>
75         </rendeles>
76         <rendeles R_ID="5" V_IDREF="1" E_IDREF="3">
77             <ar>5490</ar>
78             <ido>16:32</ido>
79         </rendeles>
80     </rendeles>
81

```

```

82     <ettermek>
83         <etterm E_ID="1" F_IDREF="1">
84             <nev>Bitang Joe Burgerbár</nev>
85             <nyitvatartas>10:00-23:00</nyitvatartas>
86             <elerhetoseg>
87                 <weboldal>www.bitangjoe.hu</weboldal>
88                 <telefonszam>+4630367876</telefonszam>
89             </elerhetoseg>
90         </etterm>
91         <etterm E_ID="2" F_IDREF="3">
92             <nev>Pizza, kávé világbéke</nev>
93             <nyitvatartas>8:00-22:00</nyitvatartas>
94             <elerhetoseg>
95                 <weboldal>www.vilagbeke.hu</weboldal>
96                 <telefonszam>+4630654987</telefonszam>
97             </elerhetoseg>
98         </etterm>
99         <etterm E_ID="3" F_IDREF="2">
100             <nev>Pizza, Kávé Világbéke</nev>
101             <nyitvatartas>8:00-22:00</nyitvatartas>
102             <elerhetoseg>
103                 <weboldal>www.vilagbeke.hu</weboldal>
104                 <telefonszam>+4630654987</telefonszam>
105             </elerhetoseg>
106         </etterm>
107     </ettermek>
108

```

```
109     <futarok>
110         <futar F_ID="1" E_IDREF="2">
111             <nev>Kiss Miklós</nev>
112             <telefonszam>+36207652245</telefonszam>
113         </futar>
114         <futar F_ID="2" E_IDREF="1">
115             <nev>Nagy Máté</nev>
116             <telefonszam>+36303873378</telefonszam>
117         </futar>
118         <futar F_ID="3" E_IDREF="3">
119             <nev>Kovács András</nev>
120             <telefonszam>+36303873378</telefonszam>
121         </futar>
122         <futar F_ID="4" E_IDREF="3">
123             <nev>Takács Zoltán</nev>
124             <telefonszam>+36207652245</telefonszam>
125         </futar>
126         <futar F_ID="5" E_IDREF="2">
127             <nev>Nagy Ervin</nev>
128             <telefonszam>+36303873378</telefonszam>
129         </futar>
130         <futar F_ID="6" E_IDREF="1">
131             <nev>Ötvös Ádám</nev>
132             <telefonszam>+36303873378</telefonszam>
133         </futar>
134     </futarok>
135
```

```
136     <dolgozni>
137         <dolgozik E_IDREF="1" F_IDREF="1">
138             <mikor>16:00-23:00</mikor>
139             <mivel>Fiat Punto</mivel>
140         </dolgozik>
141         <dolgozik E_IDREF="2" F_IDREF="2">
142             <mikor>12:00-22:00</mikor>
143             <mivel>Suzuki Swift</mivel>
144         </dolgozik>
145         <dolgozik E_IDREF="1" F_IDREF="3">
146             <mikor>10:00-18:00</mikor>
147             <mivel>Skoda Fabia</mivel>
148         </dolgozik>
149         <dolgozik E_IDREF="3" F_IDREF="4">
150             <mikor>11:00-23:00</mikor>
151             <mivel>Suzuki Ignis</mivel>
152         </dolgozik>
153         <dolgozik E_IDREF="1" F_IDREF="5">
154             <mikor>14:00-22:00</mikor>
155             <mivel>Suzuki Ignis</mivel>
156         </dolgozik>
157         <dolgozik E_IDREF="2" F_IDREF="6">
158             <mikor>15:00-20:00</mikor>
159             <mivel>Suzuki Swift</mivel>
160         </dolgozik>
161     </dolgozni>
162 </etelrendeles>
```

Séma kialakítása az XML alapján:

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
3
4
5      <xs:element name="kartyaszam" type="xs:integer"/>
6      <xs:element name="lejarat" type="xs:string"/>
7      <xs:element name="bank" type="xs:string"/>
8      <xs:element name="tipus" type="xs:string"/>
9      <xs:element name="nev" type="xs:string"/>
10     <xs:element name="iranyitoszam" type="xs:integer"/>
11     <xs:element name="varos" type="xs:string"/>
12     <xs:element name="utca" type="xs:string"/>
13     <xs:element name="hazszam" type="xs:integer"/>
14     <xs:element name="telefonszam" type="xs:integer"/>
15     <xs:element name="ido" type="xs:string"/>
16     <xs:element name="ar" type="xs:integer"/>
17     <xs:element name="nyitvatartas" type="xs:string"/>
18     <xs:element name="weboldal" type="xs:string"/>
19     <xs:element name="mikor" type="xs:string"/>
20     <xs:element name="mivel" type="xs:string"/>
21
22
23     <xs:attribute name="K_ID" type="xs:integer"/>
24     <xs:attribute name="K_IDREF" type="xs:integer"/>
25     <xs:attribute name="V_ID" type="xs:integer"/>
26     <xs:attribute name="V_IDREF" type="xs:integer"/>
27     <xs:attribute name="R_ID" type="xs:integer"/>
28     <xs:attribute name="R_IDREF" type="xs:integer"/>
29     <xs:attribute name="E_ID" type="xs:integer"/>
30     <xs:attribute name="E_IDREF" type="xs:integer"/>
31     <xs:attribute name="F_ID" type="xs:integer"/>
32     <xs:attribute name="F_IDREF" type="xs:integer"/>
33
```



```
34 <xs:complexType name="cimtype">
35   <xs:sequence>
36     <xs:element ref="iranyitoszam"/>
37     <xs:element ref="varos"/>
38     <xs:element ref="utca"/>
39     <xs:element ref="hazszam"/>
40   </xs:sequence>
41 </xs:complexType>
42
43 <xs:complexType name="elerhetosegtype">
44   <xs:sequence>
45     <xs:element ref="weboldal"/>
46     <xs:element ref="telefonszam"/>
47   </xs:sequence>
48 </xs:complexType>
49
50 <xs:complexType name="dolgoziktype">
51   <xs:sequence>
52     <xs:element ref="mikor"/>
53     <xs:element ref="mivel"/>
54   </xs:sequence>
55   <xs:attribute ref="E_IDREF" use="required"/>
56   <xs:attribute ref="F_IDREF" use="required"/>
57 </xs:complexType>
58
59 <xs:complexType name="bankkartyatype">
60   <xs:sequence>
61     <xs:element ref="kartyaszam"/>
62     <xs:element ref="lejarat"/>
63     <xs:element ref="bank"/>
64     <xs:element ref="tipus"/>
65   </xs:sequence>
66   <xs:attribute ref="K_ID" use="required"/>
67 </xs:complexType>
68
```

```

69     <xs:complexType name="vevotype">
70         <xs:sequence>
71             <xs:element ref="nev"/>
72             <xs:element name="cim" type="cimtype" maxOccurs="1"/>
73             <xs:element ref="telefonszam"/>
74         </xs:sequence>
75         <xs:attribute ref="V_ID" use="required"/>
76         <xs:attribute ref="K_IDREF" use="required"/>
77         <xs:attribute ref="R_IDREF" use="required"/>
78     </xs:complexType>
79
80     <xs:complexType name="rendelestype">
81         <xs:sequence>
82             <xs:element ref="ar"/>
83             <xs:element ref="ido"/>
84         </xs:sequence>
85         <xs:attribute ref="R_ID" use="required"/>
86         <xs:attribute ref="V_IDREF" use="required"/>
87         <xs:attribute ref="E_IDREF" use="required"/>
88     </xs:complexType>
89
90     <xs:complexType name="etteremtype">
91         <xs:sequence>
92             <xs:element ref="nev"/>
93             <xs:element ref="nyitvatartas"/>
94             <xs:element name="elerhetoseg" type="elerhetosegtype" maxOccurs="1"/>
95         </xs:sequence>
96         <xs:attribute ref="E_ID" use="required"/>
97         <xs:attribute ref="F_IDREF" use="required"/>
98     </xs:complexType>
99

```

```

100    <xs:complexType name="futartype">
101        <xs:sequence>
102            <xs:element ref="nev"/>
103            <xs:element ref="telefonszam"/>
104        </xs:sequence>
105        <xs:attribute ref="F_ID" use="required"/>
106        <xs:attribute ref="E_IDREF" use="required"/>
107    </xs:complexType>
108
109
110    <xs:complexType name="bankkartyaktype">
111        <xs:sequence>
112            <xs:element name="bankkartya" type="bankkartyatype" maxOccurs="unbounded"/>
113        </xs:sequence>
114    </xs:complexType>
115
116    <xs:complexType name="vevoktype">
117        <xs:sequence>
118            <xs:element name="vevo" type="vevotype" maxOccurs="unbounded"/>
119        </xs:sequence>
120    </xs:complexType>
121
122    <xs:complexType name="rendelesektype">
123        <xs:sequence>
124            <xs:element name="rendeles" type="rendelestype" maxOccurs="unbounded"/>
125        </xs:sequence>
126    </xs:complexType>
127
128    <xs:complexType name="ettermektype">
129        <xs:sequence>
130            <xs:element name="etterem" type="etteremtype" maxOccurs="unbounded"/>
131        </xs:sequence>
132    </xs:complexType>
133

```

```
134 <xs:complexType name="futaroktype">
135 |   <xs:sequence>
136 |     <xs:element name="futar" type="futartype" maxOccurs="unbounded"/>
137 |   </xs:sequence>
138 </xs:complexType>
139
140 <xs:complexType name="dolgoznitype">
141 |   <xs:sequence>
142 |     <xs:element name="dolgozik" type="dolgoziktype" maxOccurs="unbounded"/>
143 |   </xs:sequence>
144 </xs:complexType>
145
146
147 <xs:element name="etelrendeles">
148 |   <xs:complexType>
149 |     <xs:sequence>
150 |       <xs:element name="bankkartyak" type="bankkartyaktype"/>
151 |       <xs:element name="vevok" type="vevoktype"/>
152 |       <xs:element name="rendelessek" type="rendelesektype"/>
153 |       <xs:element name="ettermek" type="ettermektype"/>
154 |       <xs:element name="futarok" type="futaroktype"/>
155 |       <xs:element name="dolgozni" type="dolgoznitype"/>
156 |     </xs:sequence>
157 |   </xs:complexType>
158
159 |   <xs:key name="K_ID">
160 |     <xs:selector xpath="bankkartyak/bankkartya"/>
161 |     <xs:field xpath="@K_ID"/>
162 |   </xs:key>
163
164 |   <xs:key name="V_ID">
165 |     <xs:selector xpath="vevok/vevo"/>
166 |     <xs:field xpath="@V_ID"/>
167 |   </xs:key>
168
```

```

169     <xs:key name="R_ID">
170         <xs:selector xpath="rendelesek/rendeles"/>
171         <xs:field xpath="@R_ID"/>
172     </xs:key>
173
174     <xs:key name="E_ID">
175         <xs:selector xpath="ettermek/etterem"/>
176         <xs:field xpath="@E_ID"/>
177     </xs:key>
178
179     <xs:key name="F_ID">
180         <xs:selector xpath="futarok/futar"/>
181         <xs:field xpath="@F_ID"/>
182     </xs:key>
183
184     <xs:keyref name="VevoFK1" refer="V_ID">
185         <xs:selector xpath="bankkartyak/bankkartya"/>
186         <xs:field xpath="@V_IDREF"/>
187     </xs:keyref>
188     <xs:keyref name="VevoFK2" refer="V_ID">
189         <xs:selector xpath="rendelesek/rendeles"/>
190         <xs:field xpath="@V_IDREF"/>
191     </xs:keyref>
192     <xs:keyref name="RendelesFK1" refer="R_ID">
193         <xs:selector xpath="bankkartyak/bankkartya"/>
194         <xs:field xpath="@R_IDREF"/>
195     </xs:keyref>
196     <xs:keyref name="RendelesFK2" refer="R_ID">
197         <xs:selector xpath="vevok/vevo"/>
198         <xs:field xpath="@R_IDREF"/>
199     </xs:keyref>
200     <xs:keyref name="EtteremFK1" refer="V_ID">
201         <xs:selector xpath="rendelesek/rendeles"/>
202         <xs:field xpath="@E_IDREF"/>
203     </xs:keyref>

```

```

204     <xs:keyref name="EtteremFK2" refer="V_ID">
205         <xs:selector xpath="futarok/futar"/>
206         <xs:field xpath="@E_IDREF"/>
207     </xs:keyref>
208     <xs:keyref name="DolgozikFK1" refer="V_ID">
209         <xs:selector xpath="futarok/futar"/>
210         <xs:field xpath="@D_IDREF"/>
211     </xs:keyref>
212     <xs:keyref name="DolgozikFK2" refer="V_ID">
213         <xs:selector xpath="ettermek/etterem"/>
214         <xs:field xpath="@D_IDREF"/>
215     </xs:keyref>
216 </xs:element>
217
218
219 </xs:schema>
220

```

Java DOM programok:

DOMRead

A Java program beimportálja az XML file adatait, majd ezután kiírja azokat a konzolra.

```
1  package hu.domparsa.ICDOQV.DOMRead;
2
3  import ...
16
17  public class Domread {
18  public static void main(String[] args) {
19
20      //Új DocumentBuilder létrehozása
21      //XML fájlból dokumentum létrehozása
22      File xmlFile = new File( pathname: "XML_ICDOQV.xml");
23      Document doc = null;
24      doc = introduceFile(doc, xmlFile);
25
26      //Teszteljük, hogy lefut-e, ha igen kiírjuk
27      if (doc != null) {
28          doc.getDocumentElement().normalize();
29          System.out.println("Root element: " + doc.getDocumentElement().getNodeName());
30      } else {
31          System.out.println("A doc null érték!");
32      }
33      NodeList myList = doc.getDocumentElement().getChildNodes();
34      String indent = "";
35      listData(myList, indent);
36
37  }
39
40  public static Document introduceFile(Document doc, File xmlFile) {
41      try {
42          DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
43          DocumentBuilder dBuilder = factory.newDocumentBuilder();
44          doc = dBuilder.parse(xmlFile);
45      } catch (ParserConfigurationException | SAXException | IOException ex) {
46          Logger.getLogger(Domread.class.getName()).log(Level.SEVERE, msg: null, ex);
47      }
48      return doc;
49  }
50
51  public static void listData(NodeList thisList, String indent) {
52      indent += "\t";
53      if (thisList != null) {
54          for (int i = 0; i < thisList.getLength(); i++) {
55              Node thisNode = thisList.item(i);
56              if (thisNode.getNodeType() == Node.ELEMENT_NODE && !thisNode.getTextContent().trim().equals("")) {
57                  System.out.println(indent + "{" + thisNode.getNodeName() + "}:");
58
59                  NodeList newList = thisNode.getChildNodes();
60                  listData(newList, indent);
61              } else if (thisNode instanceof Text) {
62                  String value = thisNode.getNodeValue().trim();
```

```

63         if (value.equals("")) {
64             continue;
65         }
66         System.out.println(indent + "-" + thisNode.getTextContent() + "-");
67     }
68 }
69 }
70 }
71 }
72

```

Kimenet:

```

"C:\Program Files\Eclipse Adoptium\jdk-17.0.12-hotspot\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2021.3\lib\idea_rt.jar
Root element: etelrendeles
{bankkartyak}:
  {bankkartya}:
    {kartyaszam}:
      -5321804176549876-
    {lejarat}:
      -08/23-
    {bank}:
      -OTP Bank-
    {tipus}:
      -Junior-
  {bankkartya}:
    {kartyaszam}:
      -6543778237610094-
    {lejarat}:
      -11/22-
    {bank}:
      -Erste Bank-
    {tipus}:
      -Jump-
  {bankkartya}:
    {kartyaszam}:
      -5321804198812560-

```

```

    {kartyaszam}:
      -5321804198812560-
    {lejarat}:
      -04/23-
    {bank}:
      -OTP Bank-
    {tipus}:
      -Hitel-
  {vevok}:
    {vevo}:
      {nev}:
        -Sirankó Boldizsár-
      {cim}:
        {iranyitoszam}:
          -9432-
        {varos}:
          -Kaposvár-
        {utca}:
          -Dózsa György utca-
        {hazszam}:
          -32-
      {telefonszam}:
        -06303658765-

```

```
{vevo}:
  {nev}:
    -Timkó Attila-
  {cim}:
    {iranyitoszam}:
      -4500-
    {varos}:
      -Ózd-
    {utca}:
      -Bánya út-
    {hazszam}:
      -45-
    {telefonszam}:
      -06307629813-
{vevo}:
  {nev}:
    -Szabó Zoltán-
  {cim}:
    {iranyitoszam}:
      -7640-
    {varos}:
      -Hajdúnánás-
    {utca}:
      -Mező utca-
```

```
    {hazszam}:
      -2-
    {telefonszam}:
      -06709463371-
{rendelesek}:
  {rendeles}:
    {ar}:
      -2690-
    {ido}:
      -11:32-
  {rendeles}:
    {ar}:
      -9990-
    {ido}:
      -13:52-
  {rendeles}:
    {ar}:
      -3450-
    {ido}:
      -14:14-
  {rendeles}:
    {ar}:
      -10550-
```

```
    {ido}:
      -15:22-
  {rendeles}:
    {ar}:
      -5490-
    {ido}:
      -16:32-
{ettermek}:
  {etterem}:
    {nev}:
      -Bitang Joe Burgerbár-
    {nyitvatartas}:
      -10:00-23:00-
    {elerhetoseg}:
      {weboldal}:
        -www.bitangjoe.hu-
      {telefonszam}:
        -+4630367876-
  {etterem}:
    {nev}:
      -Pizza, Kávé világbéke-
    {nyitvatartas}:
      -10550-
```

```
{nyitvatartas}:
  -8:00-22:00-
{elerhetoseg}:
  {weboldal}:
    -www.vilagbeke.hu-
  {telefonszam}:
    -+4630654987-
{etterem}:
  {nev}:
    -Pizza, Kávé Világbéke-
  {nyitvatartas}:
    -8:00-22:00-
  {elerhetoseg}:
    {weboldal}:
      -www.vilagbeke.hu-
    {telefonszam}:
      -+4630654987-
{futarok}:
  {futar}:
    {nev}:
      -Kiss Miklós-
    {telefonszam}:
      -+36207652245-
```

```
{futar}:
  {nev}:
    -Nagy Máté-
  {telefonszam}:
    -+36303873378-
{futar}:
  {nev}:
    -Kovács András-
  {telefonszam}:
    -+36303873378-
{futar}:
  {nev}:
    -Takács Zoltán-
  {telefonszam}:
    -+36207652245-
{futar}:
  {nev}:
    -Nagy Ervin-
  {telefonszam}:
    -+36303873378-
```

```
{futar}:
  {nev}:
    -Ötvös Ádám-
  {telefonszam}:
    -+36303873378-
```

Process finished with exit code 0

DOMModify

A Java program módosítja a rendelések értékét, csökkenti azokat 1000 forinttal, majd az így módosított adatokat írja ki a konzolra.

```
package hu.domparsa.ICDOQV.DOMModify;

import ...

public class Dommodify {

    public static void main(String[] args) {

        //Uj DocumentBuilder létrehozasa
        //XML fajlbol dokumentum létrehozasa
        File xmlFile = new File( pathname: "XML_ICDOQV.xml");
        Document doc = null;
        doc = introduceFile(doc, xmlFile);

        //Teszteljuk, hogy lefut-e, ha igen kiirjuk
        if (doc != null) {
            doc.getDocumentElement().normalize();
            System.out.println("Root element: " + doc.getDocumentElement().getNodeName());
        } else {
            System.out.println("A doc null ertekek!");
        }
        NodeList myList = doc.getDocumentElement().getChildNodes();
        NodeList modifyList = doc.getDocumentElement().getElementsByTagName("ar");

        //Modositas
        modifyData(modifyList);
        String indent = "";

        listData(myList, indent);
    }

    public static Document introduceFile(Document doc, File xmlFile) {
        try {
            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
            DocumentBuilder dBuilder = factory.newDocumentBuilder();
            doc = dBuilder.parse(xmlFile);
        } catch (ParserConfigurationException | SAXException | IOException ex) {
            Logger.getLogger(Dommodify.class.getName()).log(Level.SEVERE, msg: null, ex);
        }
        return doc;
    }
}
```

```

57     public static void listData(NodeList thisList, String indent) {
58         indent+="\t";
59         if (thisList != null) {
60             for (int i = 0; i < thisList.getLength(); i++) {
61                 Node thisNode = thisList.item(i);
62                 if (thisNode.getNodeType() == Node.ELEMENT_NODE && !thisNode.getTextContent().trim().equals("")) {
63                     System.out.println(indent+"{"+thisNode.getNodeName()+"}:"");
64
65                     NodeList newList = thisNode.getChildNodes();
66                     listData(newList, indent);
67                 } else if (thisNode instanceof Text) {
68                     String value = thisNode.getNodeValue().trim();
69                     if (value.equals("")) {
70                         continue;
71                     }
72                     System.out.println(indent+"-"+thisNode.getTextContent()+"-");
73                 }
74             }
75         }
76     }

```

```

77     //ar csokkentese 1000 forinttal
78     public static void modifyData(NodeList thisList){
79         if(thisList!=null){
80             for (int i = 0; i < thisList.getLength(); i++) {
81                 int ar = Integer.parseInt(thisList.item(i).getTextContent());
82                 ar-=1000;
83                 Integer myAr = ar;
84                 thisList.item(i).setTextContent(myAr.toString());
85             }
86         }
87     }
88 }
89

```

Kimenet (módosított rész):

```

{rendeles}:
  {ar}:
    -1690-
  {ido}:
    -11:32-
{rendeles}:
  {ar}:
    -8990-
  {ido}:
    -13:52-
{rendeles}:
  {ar}:
    -2450-
  {ido}:
    -14:14-
{rendeles}:
  {ar}:
    -9550-
  {ido}:
    -15:22-
{rendeles}:
  {ar}:
    -4490-

```

DOMQuery

A Java program azokat a rendeléseket gyűjti ki, melyek 5000 forint feletti értékűek.

```
1 package hu.domparsa.ICDQV.DOMQuery;
2
3 import ...
4
14
15 public class Domquery {
16 public static void main(String[] args) {
17
18     //Uj DocumentBuilder létrehozasa
19     //XML fajlbol dokumentum létrehozasa
20     File xmlFile = new File( pathname: "XML_ICDQV.xml");
21     Document doc = null;
22     doc = introduceFile(doc, xmlFile);
23
24     //Teszteljuk, hogy lefut-e, ha igen kiirjuk
25     if (doc != null) {
26         doc.getDocumentElement().normalize();
27         // System.out.println("Root element: " + doc.getDocumentElement().getNodeName());
28     } else {
29         System.out.println("A doc null ertekek!");
30     }
31     NodeList myList = doc.getDocumentElement().getChildNodes();
32
33     String indent = "";
34
35     //Query - rendelések
36     NodeList queryList = doc.getDocumentElement().getElementsByTagName("rendeles");
37     String keresett = null;
38
39     //5000 forint feletti rendelések kigyujtese
40     for (int i = 0; i < queryList.getLength(); i++) {
41         NodeList query = queryList.item(i).getChildNodes();
42         for (int j = 0; j < query.getLength(); j++) {
43             if (query.item(j).getNodeName() == "ar" && Integer.parseInt(query.item(j).getTextContent())>5000) {
44                 System.out.println("{rendeles}");
45                 listData(queryList.item(i).getChildNodes(), indent: "");
46             }
47         }
48     }
49
50
51 }
52
```

```

53 @ public static String getAttribute(Node myNode, String ID) {
54     NamedNodeMap thisMap = myNode.getAttributes();
55     for (int i = 0; i < thisMap.getLength(); i++) {
56         if (thisMap.item(i).getNodeName().equals(ID)) {
57             return thisMap.item(i).getTextContent();
58         }
59     }
60     return "";
61 }
62
63
64 @ public static String getParentAttribute(Node myNode, String ID) {
65     NamedNodeMap thisMap = myNode.getParentNode().getAttributes();
66     for (int i = 0; i < thisMap.getLength(); i++) {
67         if (thisMap.item(i).getNodeName().equals(ID)) {
68             return thisMap.item(i).getTextContent();
69         }
70     }
71     return "";
72 }
73
74

```

```

75 public static Document introduceFile(Document doc, File xmlFile) {
76     try {
77         DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
78         DocumentBuilder dBuilder = factory.newDocumentBuilder();
79         doc = dBuilder.parse(xmlFile);
80     } catch (ParserConfigurationException | SAXException | IOException ex) {
81         Logger.getLogger(Domquery.class.getName()).log(Level.SEVERE, msg: null, ex);
82     }
83     return doc;
84 }
85
86
87 public static void listData(NodeList thisList, String indent) {
88     indent += "\t";
89     if (thisList != null) {
90         for (int i = 0; i < thisList.getLength(); i++) {
91             Node thisNode = thisList.item(i);
92             if (thisNode.getNodeType() == Node.ELEMENT_NODE && !thisNode.getTextContent().trim().equals("")) {
93                 System.out.println(indent + "{" + thisNode.getNodeName() + "}:" );
94
95                 NodeList newList = thisNode.getChildNodes();
96                 listData(newList, indent);
97             } else if (thisNode instanceof Text) {

```

```

98                 String value = thisNode.getNodeValue().trim();
99                 if (value.equals("")) {
100                     continue;
101                 }
102                 System.out.println(indent + "-" + thisNode.getTextContent() + "-");
103             }
104         }
105     }
106 }
107
108

```

Kimenet:

```
"C:\Program Files\Eclipse Adoptium\jdk-17.0.12-hotspot\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2021.3\lib\idea_rt.jar
{rendeles}
  {ar}:
    -9990-
  {ido}:
    -13:52-
{rendeles}
  {ar}:
    -10550-
  {ido}:
    -15:22-
{rendeles}
  {ar}:
    -5490-
  {ido}:
    -16:32-

Process finished with exit code 0
```