Jegyzőkönyv
Adatkezelés XML-ben
Féléves feladat
Airbnb lakásközvetítő cég adatbázisa

Készítette: **Tamás Janka**
Neptun-kód: **GIE0EJ**
Gyakorlat: **szerda, 10:00-12:00**
Gyakorlat vezető: **Dr. Bednarik László**
Kelt: 2021.11.30.

Tartalom

Feladat leírása:

A tematika egy airbnb közvetítő cég adatbázisa, melyben egy airbnb lakás kiadásához szükséges adatok vannak nyilvántartva. Az adatbázis a lakásadatokon kívül nyilvántartja a bérlő vendégek és foglalásuk adatait, illetve a lakásokat takarító személyzetet is. Egy kiadás menete a következőképpen alakul: a vendég lefoglalja az általa kinézett lakást, majd a foglaláshoz az oldal vagy a cég egy dolgozója hozzárendel egy takarítót, aki a vendég távozása után kitakarítja a lakást.

Az ER modell egyedei és tulajdonságai:

Az egyedek a következők: Lakás, Vendég, Foglalás, Takarító

Egyedek tulajdonságai:

- **Lakás:**
  1. LakásID: elsődleges kulcs
  2. Kapukód: a lakás főkapujának a belépőkódja
  3. Cím: összetett tulajdonság, a lakás címét tárolja, azon belül irányítószám, település, utca, házszám, emelet és ajtó adatokat

- **Vendég:**
  1. VendégID: elsődleges kulcs
  2. Név: a vendég neve
  3. Telefonszám: többértékű tulajdonság, a tulajdonos telefonszáma(i), elérhetősége
  4. Útlevél száma: a vendég útlevelének az azonosítója, ezzel azonosítja magát a személy (ez lehet még személyi igazolvány és jogosítvány is)

- **Foglalás:**
  1. FoglalásID: elsődleges kulcs
  2. Fő: megadja, hogy hány főre szól a foglalás
  3. Kezdő dátum: vendég érkezésének dátuma
  4. Záró dátum: vendég távozásának dátuma
  5. Éjszakák száma: származtatott tulajdonság, a „kezdő dátumnak” a „záró dátumból” való kivonásával számítódik ki

- **Takarító:**
  1. TakarítóID: elsődleges kulcs
  2. Név: a takarító neve
  3. Személyi igazolvány szám: ez az adat azonosítja a személyt
  4. Mobilszám: a takarító elérhetősége, mobilszáma

Egyedek közötti kapcsolat

Lakás – Foglalás → 1:1 kapcsolat:
- egy lakás csak egy foglalásban szerepelhet
- egy foglaláshoz csak egy lakás tartozik

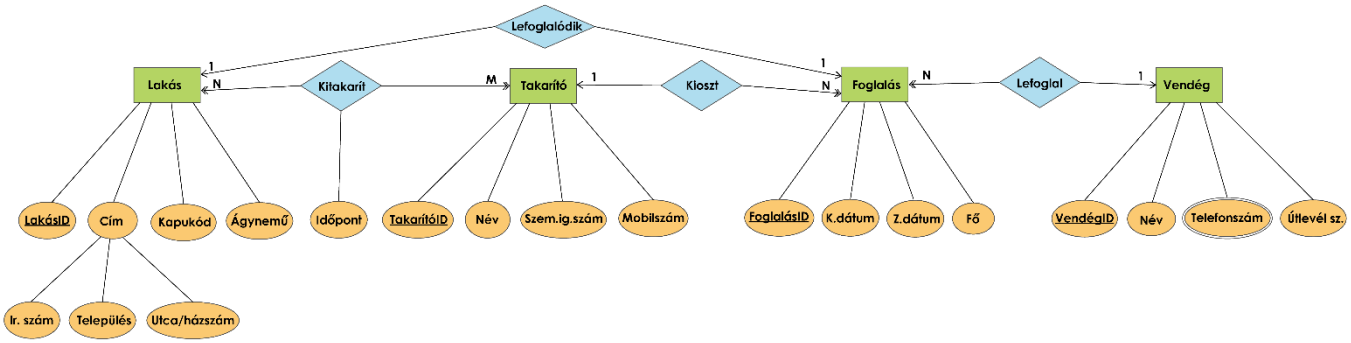
Lakás – Takarító → N:M kapcsolat:
- egy lakást több takarító takaríthat
- illetve egy takarító több lakást takaríthat

Takarító – Foglalás → 1:N kapcsolat:
- egy takarítónak több foglalás is juthat, aminek a lakását ki kell takarítania
- viszont egy foglaláshoz csak egy takarítót oszt be a rendszer

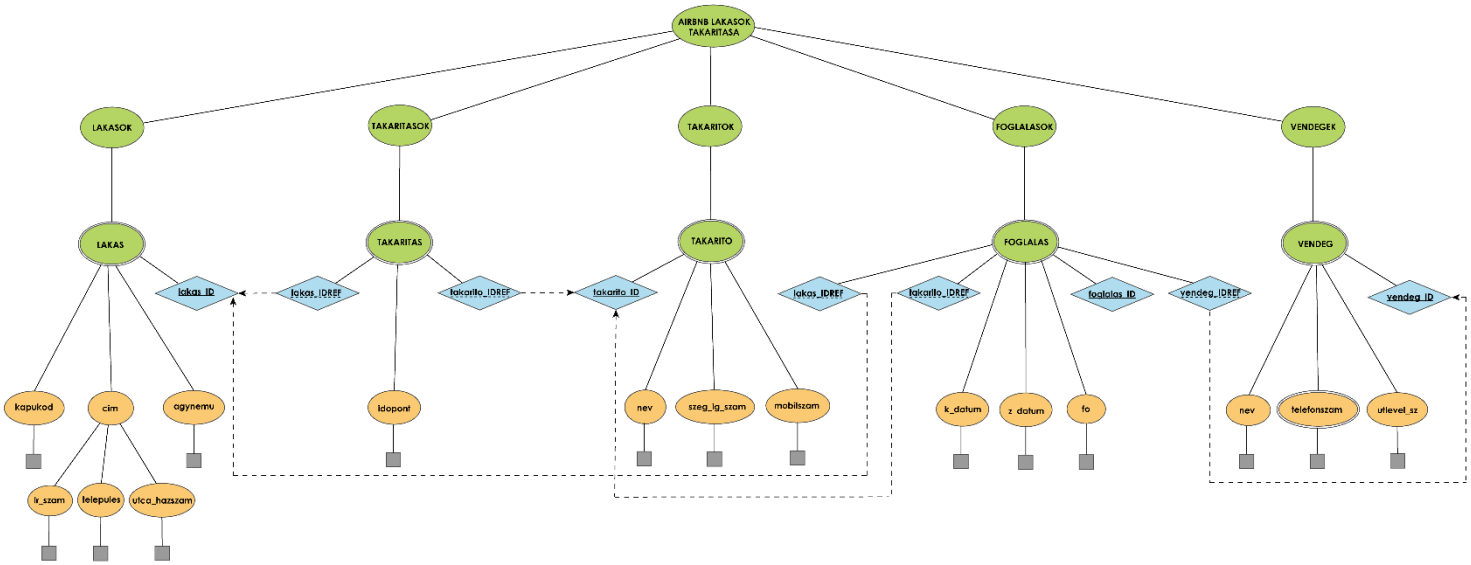
Vendég – Foglalás → 1:N kapcsolat:
- egy vendég több foglalást is végre tud hajtani
- viszont egy foglaláshoz csak egy vendég tartozhat

1. feladat

1a, ER modell



1b, XDM modell



1c, XML fájl:

```xml
<?xml version="1.0" encoding="UTF-8"?>

<airbnblakasoktakaritasa xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="XMLSchemaGIE0EJ.xsd">
    <lakasok>
        <lakas lakas_ID="1">
            <kapukod>2345</kapukod>
            <cim>
                <ir_szam>1065</ir_szam>
                <telepules>Budapest</telepules>
                <utca_hazszam>Révay utca 8</utca_hazszam>
            </cim>
            <agynemu>szimpla</agynemu>
        </lakas>
        <lakas lakas_ID="2">
            <kapukod>3456</kapukod>
            <cim>
                <ir_szam>1061</ir_szam>
                <telepules>Budapest</telepules>
                <utca_hazszam>Andrássy út 23</utca_hazszam>
            </cim>
            <agynemu>dupla</agynemu>
        </lakas>
        <lakas lakas_ID="3">
            <kapukod>4585</kapukod>
            <cim>
                <ir_szam>1051</ir_szam>
                <telepules>Budapest</telepules>
                <utca_hazszam>Zrínyi utca 4</utca_hazszam>
            </cim>
```

```
            </cim>
            <agynemu>szimpla</agynemu>
        </lakas>
    </lakasok>
    <takaritasok>
        <takaritas lakas_IDREF="1" takarito_IDREF="3">
            <idopont>2021-12-01</idopont>
        </takaritas >
        <takaritas lakas_IDREF="3" takarito_IDREF="2">
            <idopont>2021-12-04</idopont>
        </takaritas >
        <takaritas lakas_IDREF="2" takarito_IDREF="1">
            <idopont>2021-12-06</idopont>
        </takaritas >
    </takaritasok>
    <takaritok>
        <takarito takarito_ID="1">
            <nev>Kiss Mária</nev>
            <szem_ig_szam>231456PA</szem_ig_szam>
            <mobilszam>06308573245</mobilszam>
        </takarito>
        <takarito takarito_ID="2">
            <nev>Nagy Katalin</nev>
            <szem_ig_szam>463782EE</szem_ig_szam>
            <mobilszam>06204563998</mobilszam>
        </takarito>
        <takarito takarito_ID="3">
            <nev>Szabó Péter</nev>
            <szem_ig_szam>125478LD</szem_ig_szam>
            <szem_ig_szam>125478LD</szem_ig_szam>
            <mobilszam>06703456787</mobilszam>
        </takarito>
    </takaritok>
    <foglalasok>
        <foglalas foglalas_ID="1" lakas_IDREF="1" takarito_IDREF="3"  vendeg_IDREF="1">
            <k_datum>2021-11-29</k_datum>
            <z_datum>2021-12-01</z_datum>
            <fo>3</fo>
        </foglalas>
        <foglalas foglalas_ID="2" lakas_IDREF="3" takarito_IDREF="2"  vendeg_IDREF="2">
            <k_datum>2021-12-03</k_datum>
            <z_datum>2021-12-04</z_datum>
            <fo>2</fo>
        </foglalas>
        <foglalas foglalas_ID="3" lakas_IDREF="2" takarito_IDREF="1"  vendeg_IDREF="3">
            <k_datum>2021-12-04</k_datum>
            <z_datum>2021-12-06</z_datum>
            <fo>2</fo>
        </foglalas>
    </foglalasok>
    <vendegek>
        <vendeg vendeg_ID="1">
            <nev>John Smith</nev>
            <telefonszam>+223456785</telefonszam>
            <telefonszam>06304567855</telefonszam>
            <utlevel_sz>234567AA</utlevel_sz>
        </vendeg>
        <vendeg vendeg_ID="2">
            <nev>Nagy Tamás</nev>
            <telefonszam>06702874956</telefonszam>
            <utlevel_sz>475839BD</utlevel_sz>
        </vendeg>
        <vendeg vendeg_ID="3">
            <nev>Kovács Piroska</nev>
            <telefonszam>06303455443</telefonszam>
            <telefonszam>061223678</telefonszam>
            <utlevel_sz>347656PL</utlevel_sz>
        </vendeg>
    </vendegek>
</airbnblakasoktakaritasa>
```

1d, XML Schema

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

<xs:element name="kapukod" type="xs:integer"/>
<xs:element name="ir_szam" type="xs:integer"/>
<xs:element name="telepules" type="xs:string"/>
<xs:element name="utca_hazszam" type="xs:string"/>
<xs:element name="agynemu" type="xs:string"/>
<xs:element name="idopont" type="xs:date"/>
<xs:element name="nev" type="xs:string"/>
<xs:element name="szem_ig_szam" type="xs:string"/>
<xs:element name="mobilszam" type="xs:string"/>
<xs:element name="k_datum" type="xs:date"/>
<xs:element name="z_datum" type="xs:date"/>
<xs:element name="fo" type="xs:integer"/>
<xs:element name="utlevel_sz" type="xs:string"/>
<xs:element name="telefonszam" type="xs:string"/>

<xs:attribute name="lakas_ID" type="xs:integer"/>
<xs:attribute name="takarito_ID" type="xs:integer"/>
<xs:attribute name="lakas_IDREF" type="xs:integer"/>
<xs:attribute name="takarito_IDREF" type="xs:integer"/>
<xs:attribute name="foglalas_ID" type="xs:integer"/>
<xs:attribute name="vendeg_IDREF" type="xs:integer"/>
<xs:attribute name="vendeg_ID" type="xs:integer"/>

<xs:complexType name="cim_tipus">
```

```xml
<xs:complexType name="cim_tipus">
    <xs:sequence>
        <xs:element ref="ir_szam"/>
        <xs:element ref="telepules"/>
        <xs:element ref="utca_hazszam"/>
    </xs:sequence>
</xs:complexType>


<xs:complexType name="lakasok_tipus">
    <xs:sequence>
        <xs:element name="lakas" type="lakas_tipus" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="takaritasok_tipus">
    <xs:sequence>
        <xs:element name="takaritas" type="takaritas_tipus" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="takaritok_tipus">
    <xs:sequence>
        <xs:element name="takarito" type="takarito_tipus" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="foglalasok_tipus">
```

```
<xs:complexType name="foglalasok_tipus">
    <xs:sequence>
        <xs:element name="foglalas" type="foglalas_tipus" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="vendegek_tipus">
    <xs:sequence>
        <xs:element name="vendeg" type="vendeg_tipus" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="vendeg_tipus">
    <xs:sequence>
        <xs:element ref="nev"/>
        <xs:element ref="telefonszam" maxOccurs="unbounded"  />
        <xs:element ref="utlevel_sz"/>
    </xs:sequence>
    <xs:attribute ref="vendeg_ID" use="required"/>
</xs:complexType>

<xs:complexType name="foglalas_tipus">
    <xs:sequence>
        <xs:element ref="k_datum" />
        <xs:element ref="z_datum" />
        <xs:element ref="fo" />
    </xs:sequence>
    <xs:attribute ref="foglalas_ID" use="required"/>
    <xs:attribute ref="lakas_IDREF" use="required"/>
```

```
    <xs:attribute ref="lakas_IDREF" use="required"/>
    <xs:attribute ref="takarito_IDREF" use="required"/>
    <xs:attribute ref="vendeg_IDREF" use="required"/>
</xs:complexType>

<xs:complexType name="takarito_tipus">
    <xs:sequence>
        <xs:element ref="nev" />
        <xs:element ref="szem_ig_szam" />
        <xs:element ref="mobilszam" />
    </xs:sequence>
    <xs:attribute ref="takarito_ID" use="required"/>
</xs:complexType>

<xs:complexType name="takaritas_tipus">
    <xs:sequence>
        <xs:element ref="idopont" />
    </xs:sequence>
    <xs:attribute ref="takarito_IDREF" use="required"/>
    <xs:attribute ref="lakas_IDREF" use="required"/>
</xs:complexType>

<xs:complexType name="lakas_tipus">
    <xs:sequence>
        <xs:element ref="kapukod" />
        <xs:element name="cim" type="cim_tipus" />
        <xs:element ref="agynemu" />
    </xs:sequence>
    <xs:attribute ref="lakas_ID" use="required"/>
</xs:complexType>


<xs:element name="airbnblakasoktakaritasa">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="lakasok" type="lakasok_tipus"/>
            <xs:element name="takaritasok" type="takaritasok_tipus"/>
            <xs:element name="takaritok" type="takaritok_tipus"/>
            <xs:element name="foglalasok" type="foglalasok_tipus"/>
            <xs:element name="vendegek" type="vendegek_tipus"/>
        </xs:sequence>
    </xs:complexType>

    <xs:key name="lakas_PK">
        <xs:selector xpath="lakasok/lakas"/>
        <xs:field xpath="@lakas_ID"/>
    </xs:key>
    <xs:key name="takarito_PK">
        <xs:selector xpath="takaritok/takarito"/>
        <xs:field xpath="@takarito_ID"/>
    </xs:key>
    <xs:key name="foglalas_PK">
        <xs:selector xpath="foglalasok/foglalas"/>
        <xs:field xpath="@foglalas_ID"/>
    </xs:key>
    <xs:key name="vendeg_PK">
```

```
        <xs:selector xpath="vendegek/vendeg"/>
        <xs:field xpath="@vendeg_ID"/>
    </xs:key>

    <xs:keyref name="t_lakas_FK" refer="lakas_PK">
        <xs:selector xpath="takaritasok/takaritas"/>
        <xs:field xpath="@lakas_IDREF"/>
    </xs:keyref>
    <xs:keyref name="t_takarito_FK" refer="takarito_PK">
        <xs:selector xpath="takaritasok/takaritas"/>
        <xs:field xpath="@takarito_IDREF"/>
    </xs:keyref>
    <xs:keyref name="f_lakas_FK" refer="lakas_PK">
        <xs:selector xpath="foglalasok/foglalas"/>
        <xs:field xpath="@lakas_IDREF"/>
    </xs:keyref>
    <xs:keyref name="f_takarito_FK" refer="takarito_PK">
        <xs:selector xpath="foglalasok/foglalas"/>
        <xs:field xpath="@takarito_IDREF"/>
    </xs:keyref>
    <xs:keyref name="vendeg_FK" refer="vendeg_PK">
        <xs:selector xpath="foglalasok/foglalas"/>
        <xs:field xpath="@vendeg_IDREF"/>
    </xs:keyref>
</xs:element>
</xs:schema>
```

2. feladat

2a, Read fájl

```
package hu.domparse.gie0ej;

import java.io.File;


public class DOMReadGIE0EJ {

    public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException {


        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = factory.newDocumentBuilder();


        File xmlFile = new File("src/XMLGIE0EJ.xml");
        Document doc = dBuilder.parse(xmlFile);

        doc.getDocumentElement().normalize();

        System.out.println("Root elem: " + doc.getDocumentElement().getNodeName());


        NodeList nList = doc.getElementsByTagName("lakas");

        for (int i = 0; i < nList.getLength(); i++) {


            Node nNode = nList.item(i);
            System.out.println("\nKiválasztott elem: " + nNode.getNodeName());
```

```
42
43              if (nNode.getNodeType() == Node.ELEMENT_NODE) {
44                  Element elem = (Element) nNode;
45
46                  String lakas_ID = elem.getAttribute("lakas_ID");
47
48                  Node n1 = elem.getElementsByTagName("ir_szam").item(0);
49                  String ir_szam = n1.getTextContent();
50
51                  Node n2 = elem.getElementsByTagName("telepules").item(0);
52                  String telepules = n2.getTextContent();
53
54                  Node n3 = elem.getElementsByTagName("utca_hazszam").item(0);
55                  String hazszam = n3.getTextContent();
56
57                  Node n4 = elem.getElementsByTagName("agynemu").item(0);
58                  String agynemu = n4.getTextContent();
59
60                  Node n5 = elem.getElementsByTagName("kapukod").item(0);
61                  String kapukod = n5.getTextContent();
62
63
64                  System.out.println("Lakás azonosito: " + lakas_ID);
65                  System.out.println("Kapukód: " + kapukod);
66                  System.out.println("Irányitószám: " + ir_szam);
67                  System.out.println("Település: " + telepules);
68                  System.out.println("Utca és Házszám: " + hazszam);
69                  System.out.println("Ágynemű: " + agynemu);
70
70
71              }
72          }
73
74          nList = doc.getElementsByTagName("takaritas");
75          for (int i = 0; i < nList.getLength(); i++) {
76
77              Node nNode = nList.item(i);
78              System.out.println("\nKiválasztott elem: " + nNode.getNodeName());
79
80              if (nNode.getNodeType() == Node.ELEMENT_NODE) {
81                  Element elem = (Element) nNode;
82
83                  String lakas_IDREF = elem.getAttribute("lakas_IDREF");
84                  String takarito_IDREF = elem.getAttribute("takarito_IDREF");
85
86                  Node n1 = elem.getElementsByTagName("idopont").item(0);
87                  String idopont = n1.getTextContent();
88
89                  System.out.println("Lakas azonosito: " + lakas_IDREF);
90                  System.out.println("Idopont: " + idopont);
91                  System.out.println("Takarito azonosito: " + takarito_IDREF);
92
93              }
94          }
95
96          nList = doc.getElementsByTagName("takarito");
97          for (int i = 0; i < nList.getLength(); i++) {
98
97          for (int i = 0; i < nList.getLength(); i++) {
98
99              Node nNode = nList.item(i);
100             System.out.println("\nKiválasztott elem: " + nNode.getNodeName());
101
102             if (nNode.getNodeType() == Node.ELEMENT_NODE) {
103                 Element elem = (Element) nNode;
104
105                 String takarito_ID = elem.getAttribute("takarito_ID");
106
107                 Node n1 = elem.getElementsByTagName("nev").item(0);
108                 String nev = n1.getTextContent();
109                 Node n2 = elem.getElementsByTagName("szem_ig_szam").item(0);
110                 String szem_ig_szam = n2.getTextContent();
111                 Node n3 = elem.getElementsByTagName("mobilszam").item(0);
112                 String mobilszam = n3.getTextContent();
113
114                 System.out.println("Takarito azonosito " + takarito_ID);
115                 System.out.println("Takarito neve: " + nev);
116                 System.out.println("Szemelyi igazolvany szam: " + szem_ig_szam);
117                 System.out.println("Mobilszam: " + mobilszam);
118             }
119         }
120
121         nList = doc.getElementsByTagName("vendeg");
122         for (int i = 0; i < nList.getLength(); i++) {
123
124             Node nNode = nList.item(i);
125             System.out.println("\nKiválasztott elem: " + nNode.getNodeName());
```

```
            System.out.println("\nKiválasztott elem: " + nNode.getNodeName());

            if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                Element elem = (Element) nNode;

                String vendeg_ID = elem.getAttribute("vendeg_ID");

                Node n1 = elem.getElementsByTagName("nev").item(0);
                String nev = n1.getTextContent();

                Node n2 = elem.getElementsByTagName("telefonszam").item(0);
                String telefonszam = n2.getTextContent();

                Node n3 = elem.getElementsByTagName("utlevel_sz").item(0);
                String utlevel_sz = n3.getTextContent();

                System.out.println("Vendég azonosító: " + vendeg_ID);
                System.out.println("Vendég neve: " + nev);
                System.out.println("Telefonszám: " + telefonszam);
                System.out.println("Útlevélszám: " + utlevel_sz);

            }
        }

        nList = doc.getElementsByTagName("foglalas");
        for (int i = 0; i < nList.getLength(); i++) {

            Node nNode = nList.item(i);
            System.out.println("\nKiválasztott elem: " + nNode.getNodeName());

            if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                Element elem = (Element) nNode;

                String foglalas_ID = elem.getAttribute("foglalas_ID");
                String lakas_IDREF = elem.getAttribute("lakas_IDREF");
                String takarito_IDREF = elem.getAttribute("takarito_IDREF");
                String vendeg_IDREF = elem.getAttribute("vendeg_IDREF");


                Node n1 = elem.getElementsByTagName("k_datum").item(0);
                String k_datum = n1.getTextContent();

                Node n2 = elem.getElementsByTagName("z_datum").item(0);
                String z_datum = n2.getTextContent();

                Node n3 = elem.getElementsByTagName("fo").item(0);
                String fo = n3.getTextContent();

                System.out.println("Foglalás azonosító: " + foglalas_ID);
                System.out.println("Lakás azonosító: " + lakas_IDREF);
                System.out.println("Takarító azonosító: " + takarito_IDREF);
                System.out.println("Kezdő dátum: " + k_datum);
                System.out.println("Záró dátum: " + z_datum);
                System.out.println("Létszám: " + fo);
                System.out.println("Vendég azonosító: " + vendeg_IDREF);
            }
        }
    }
}
```

Output:

```
Root elem: airbnblakasoktakaritasa

Kiválasztott elem: lakas
Lakás azonosito: 1
Kapukód: 2345
Irányitószám: 1065
Település: Budapest
Utca és Házszám: Révay utca 8
Ágynemű: szimpla

Kiválasztott elem: lakas
Lakás azonosito: 2
Kapukód: 3456
Irányitószám: 1061
Település: Budapest
Utca és Házszám: Andrássy út 23
Ágynemű: dupla

Kiválasztott elem: lakas
Lakás azonosito: 3
Kapukód: 4585
Irányitószám: 1051
Település: Budapest
Utca és Házszám: Zrínyi utca 4
Ágynemű: szimpla
```

2b, Query fájl

1. lekérdezés: lekérdezi a dupla ágyneművel rendelkező lakások adatait

2. lekérdezés: lekérdezi a 2021-12-04-én takarítandó lakás takarítójának adatait és a várost, ahol a lakás található.

```java
package hu.domparse.gie0ej;

import java.io.File;
import java.io.IOException;
import java.text.ParseException;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;

import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;

public class DOMQueryGIE0EJ {

    public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException, ParseException {

        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = factory.newDocumentBuilder();

        File xmlFile = new File("src/XMLGIE0EJ.xml");
        Document doc = dBuilder.parse(xmlFile);

        doc.getDocumentElement().normalize();

        System.out.println("Root element: " + doc.getDocumentElement().getNodeName());


        System.out.println("\nDupla ágyneművel rendelkező lakás adatai:");

        NodeList lakaslist = doc.getElementsByTagName("lakas");

        for (int i = 0; i < lakaslist.getLength(); i++) {

            Node nNode = lakaslist.item(i);

            if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                Element elem = (Element) nNode;

                String lakas_ID = elem.getAttribute("lakas_ID");

                Node n1 = elem.getElementsByTagName("kapukod").item(0);
                String kapukod = n1.getTextContent();

                Node n2 = elem.getElementsByTagName("ir_szam").item(0);
                String ir_szam = n2.getTextContent();

                Node n3 = elem.getElementsByTagName("telepules").item(0);
                String telepules = n3.getTextContent();

                Node n4 = elem.getElementsByTagName("utca_hazszam").item(0);
                String utca_hazszam = n4.getTextContent();

                Node n5 = elem.getElementsByTagName("agynemu").item(0);
                String agynemu = n5.getTextContent();

```

```
58
59                  if(agynemu.equals("dupla")) {
60
61                      System.out.println("Lakás azonositója: " + lakas_ID);
62                      System.out.println("Kapukód: " + kapukod);
63                      System.out.println("Irányitószám: " + ir_szam);
64                      System.out.println("Település: " + telepules);
65                      System.out.println("Utca és Házszám: " + utca_hazszam);
66                      System.out.println("Ágynemű: " + agynemu);
67                   }
68              }
69          }
70
71          System.out.println("\n2021-12-04-én kiirt takaritást végző dolgozó adatai és a kitakaritandó lakás települése");
72
73          NodeList takaritolist = doc.getElementsByTagName("takarito");
74          NodeList takaritaslist = doc.getElementsByTagName("takaritas");
75
76          for (int i = 0; i < takaritolist.getLength(); i++) {
77
78              Node nNode = takaritolist.item(i);
79
80              if (nNode.getNodeType() == Node.ELEMENT_NODE) {
81                  Element elem = (Element) nNode;
82
83                  String takarito_ID = elem.getAttribute("takarito_ID");
84
85
86                  Node n1 = elem.getElementsByTagName("nev").item(0);

86                  Node n1 = elem.getElementsByTagName("nev").item(0);
87                  String nev = n1.getTextContent();
y [JavaSE-13]
90                  Node n2 = elem.getElementsByTagName("szem_ig_szam").item(0);
91                  String szem_ig_szam = n2.getTextContent();
92
93                  Node n3 = elem.getElementsByTagName("mobilszam").item(0);
94                  String mobilszam = n3.getTextContent();
95
96                   for (int j = 0; j < takaritaslist.getLength(); j++) {
97
98                      Node nNode2 = takaritaslist.item(j);
99
100                     if (nNode2.getNodeType() == Node.ELEMENT_NODE) {
101                         Element elem2 = (Element) nNode2;
102
103                         String lakas_IDREF = elem2.getAttribute("lakas_IDREF");
104                         String takarito_IDREF = elem2.getAttribute("takarito_IDREF");
105
106                         Node n4 = elem2.getElementsByTagName("idopont").item(0);
107                         String idopont = n4.getTextContent();
108
109                         if(takarito_IDREF.equals(takarito_ID) &&  idopont.equals("2021-12-04")) {
110
111                             for (int k = 0; k < lakaslist.getLength(); k++) {
112
113                                 Node nNode3 = lakaslist.item(k);
114
114                                 if (nNode2.getNodeType() == Node.ELEMENT_NODE) {
115                                     Element elem3 = (Element) nNode3;
116
117                                     String lakas_ID = elem3.getAttribute("lakas_ID");
118
119                                     Node n5 = elem3.getElementsByTagName("telepules").item(0);
120                                     String telepules = n5.getTextContent();
121
122                                     if(lakas_ID.equals(lakas_IDREF) ) {
123
124                                         System.out.println("Takaritó azonositó: " + takarito_ID);
125                                         System.out.println("Takaritó neve: " + nev);
126                                         System.out.println("Személyigazolvány szám: " + szem_ig_szam);
127                                         System.out.println("Mobilszám: " + mobilszam);
128                                         System.out.println("Település: " + telepules);
129                                     }
130
131                                 }
132                             }
133                         }
134                     }
135                 }
136             }
137         }
138     }
139 }
```

Output:

```
Root element: airbnblakasoktakaritasa

Dupla ágyneművel rendelkező lakás adatai:
Lakás azonositója: 2
Kapukód: 3456
Irányitószám: 1061
Település: Budapest
Utca és Házszám: Andrássy út 23
Ágynemű: dupla

2021-12-04-én kiirt takaritást végző dolgozó adatai és a kitakaritandó lakás települése
Takaritó azonositó: 2
Takaritó neve: Nagy Katalin
Személyigazolvány szám: 463782EE
Mobilszám: 06204563998
Település: Budapest
```

2c, Modify fájl

Módosítja Nagy Katalin takarító telefonszámát.

```java
package hu.domparse.gie0ej;

import java.io.File;

public class DOMModifyGIE0EJ {

    public static void main(String[] args) {

        try {
            File inputFile = new File("src/XMLGIE0EJ.xml");
            DocumentBuilderFactory docFactory = DocumentBuilderFactory.newInstance();
            DocumentBuilder docBuilder = docFactory.newDocumentBuilder();
            Document doc = docBuilder.parse(inputFile);


          // Nagy Katalin telefonszolgáltatót váltott és új lett a mobilszáma
            NodeList takaritoList = doc.getElementsByTagName("takarito");
            for (int i = 0; i < takaritoList.getLength(); i++) {

                Node nNode1 = takaritoList.item(i);

                if (nNode1.getNodeType() == Node.ELEMENT_NODE) {
                    Element elem = (Element) nNode1;

                    Node n1 = elem.getElementsByTagName("nev").item(0);
                    String nev = n1.getTextContent();

                    if (nev.equals("Nagy Katalin")) {

                        NodeList childNodes = nNode1.getChildNodes();
```

```
                        NodeList childNodes = nNode1.getChildNodes();
                        for (int j = 0; j < childNodes.getLength(); j++) {
                            Node childNode = childNodes.item(j);
                            if (childNode.getNodeName().equals("mobilszam")) {
                                childNode.setTextContent("06303333333");


                            }

                        }

                    }
                }
            }

            modify(doc);

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


    }

    private static void modify(Document doc) throws TransformerException {
        TransformerFactory transformerFactory = TransformerFactory.newInstance();
        Transformer transformer = transformerFactory.newTransformer();
        System.out.println("-Modified File-");
        transformer.setOutputProperty(OutputKeys.ENCODING, "UTF-8");
```

```
    private static void modify(Document doc) throws TransformerException {
        TransformerFactory transformerFactory = TransformerFactory.newInstance();
        Transformer transformer = transformerFactory.newTransformer();
        System.out.println("-Modified File-");
        transformer.setOutputProperty(OutputKeys.ENCODING, "UTF-8");
        transformer.setOutputProperty(OutputKeys.INDENT, "yes");
        transformer.setOutputProperty("{http://xml.apache.org/xslt}indent-amunt", "2");

        DOMSource source = new DOMSource(doc);

        StreamResult console = new StreamResult(System.out);

        transformer.transform(source, console);

    }
}
```

Output:

```
<takarito takarito_ID="2">

    <nev>Nagy Katalin</nev>

    <szem_ig_szam>463782EE</szem_ig_szam>

    <mobilszam>06303333333</mobilszam>

</takarito>
```