JEGYZŐKÖNYV

Modern adatbázis rendszerek MSc

2023. tavasz féléves feladat

XML, XQuery, MyBatis, MongoDB

Készítette: **Tamás Janka**

Neptunkód: **GIE0EJ**

Tartalom

XML

A feladat leírása

Adatkezelés XML-ben, Java DOM parser API alkalmazása.
Téma: Airbnb lakásfoglalások nyilvántartása
Fejlesztő környezet: Eclipse

A feladat elkészítésének lépései

1. feladat

Az egyedek és tulajdonságaik, illetve az egyedek közötti kapcsolat a következőképpen alakulnak:

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

2. feladat

Ezt követően elkészítjük hozzá az XML fájlt:

```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">
```

```
        <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>
```

3. feladat

Az XML fájlhoz elkészítjük az XML Schema-t:

```
<?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">
```

```
<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>
```

4. feladat

Ezt követően elkészítjük a DOMRead java fájlt, amely a DOM parser API alkalmazásával kiolvassa az XML fájlt. A Document Object Model csomópontokat (node) használ a HTML- vagy XML-dokumentum fa struktúraként való ábrázolására.

```java
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());


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

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

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

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

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

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

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


                System.out.println("Lakás azonosito: " + lakas_ID);
                System.out.println("Kapukód: " + kapukod);
                System.out.println("Irányitószám: " + ir_szam);
                System.out.println("Település: " + telepules);
                System.out.println("Utca és Házszám: " + hazszam);
                System.out.println("Ágynemű: " + agynemu);


            }
        }

        nList = doc.getElementsByTagName("takaritas");
        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 lakas_IDREF = elem.getAttribute("lakas_IDREF");
                String takarito_IDREF = elem.getAttribute("takarito_IDREF");

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

                System.out.println("Lakas azonosito: " + lakas_IDREF);
                System.out.println("Idopont: " + idopont);
                System.out.println("Takarito azonosito: " + takarito_IDREF);

            }
        }

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

```
        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 takarito_ID = elem.getAttribute("takarito_ID");

                Node n1 = elem.getElementsByTagName("nev").item(0);
                String nev = n1.getTextContent();
                Node n2 = elem.getElementsByTagName("szem_ig_szam").item(0);
                String szem_ig_szam = n2.getTextContent();
                Node n3 = elem.getElementsByTagName("mobilszam").item(0);
                String mobilszam = n3.getTextContent();

                System.out.println("Takarito azonosito " + takarito_ID);
                System.out.println("Takarito neve: " + nev);
                System.out.println("Szemelyi igazolvany szam: " + szem_ig_szam);
                System.out.println("Mobilszam: " + mobilszam);
            }
        }

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

            Node nNode = nList.item(i);
            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);
            }
        }
    }
}
```

5. feladat

A lekérdezésekhez elkészítjük a DOMQuery java fájlt, mely az alábbi két lekérdezést hajtja végre:

1. lekérdezi a dupla ágyneművel rendelkező lakások adatait
2. 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) {

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  }
```

6. feladat

Az XML fájl módosításához elkészítjük a DOMModify java fájlt, amellyel a Nagy Katalin nevű takarító telefonszámát módosítjuk.

```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");
        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);

    }
}
```

A futtatás eredménye:

1. futtatás

A DOMRead futattásának eredménye a következő:

```
Root elem: airbnblakasoktakaritasa

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

Kiválasztott elem: lakas
Lakás azonosító: 2
Kapukód: 3456
Irányitószám: 1061
Település: Budapest
Utca és házszám: Andrássy út 23
Ágynemű: dupla
```

…

```
Kiválasztott elem: takaritas
Lakas azonosito: 1
Idopont: 2021-12-01
Takarito azonosito: 3

Kiválasztott elem: takaritas
Lakas azonosito: 3
Idopont: 2021-12-04
Takarito azonosito: 2
```

…

```
Kiválasztott elem: takarito
Takarito azonosito 1
Takarito neve: Kiss Mária
Szemelyi igazolvany szam: 231456PA
Mobilszam: 06308573245

Kiválasztott elem: takarito
Takarito azonosito 2
Takarito neve: Nagy Katalin
Szemelyi igazolvany szam: 463782EE
Mobilszam: 06204563998
```

…

```
Kiválasztott elem: vendeg
Vendég azonosító: 1
Vendég neve: John Smith
Telefonszám: +223456785
Útlevélszám: 234567AA

Kiválasztott elem: vendeg
Vendég azonosító: 2
Vendég neve: Nagy Tamás
Telefonszám: 06702874956
Útlevélszám: 475839BD
```

…

```
Kiválasztott elem: foglalas
Foglalás azonosító: 1
Lakás azonosító: 1
Takarító azonosító: 3
Kezdő dátum: 2021-11-29
Záró dátum: 2021-12-01
Létszám: 3
Vendég azonosító: 1

Kiválasztott elem: foglalas
Foglalás azonosító: 2
Lakás azonosító: 3
Takarító azonosító: 2
Kezdő dátum: 2021-12-03
Záró dátum: 2021-12-04
Létszám: 2
Vendég azonosító: 2
```

…

2. futtatás

A DOMQuery futtatásának eredménye a következő:

```
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
```

3. futtatás

A DOMModify futtatásának az eredménye:

```
<takarito takarito_ID="2">

    <nev>Nagy Katalin</nev>

    <szem_ig_szam>463782EE</szem_ig_szam>

    <mobilszam>06303333333</mobilszam>

</takarito>
```

XQuery

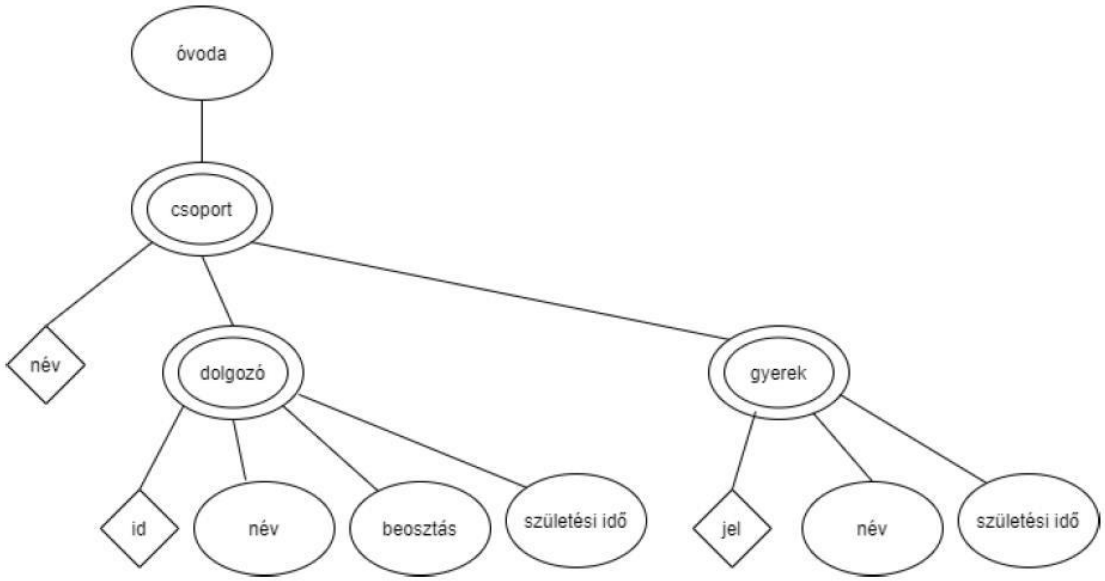
A feladat leírása

eXistDB használata, programozása, XQuery használata.
Téma: óvoda
Fejlesztő-környezet: Eclipse, eXistDB

A feladat elkészítése és a futtatási eredmények

1. feladat

Az alábbi XDM modell alapján elkészítjük az XML dokumentumot.



GIE0EJ_ovoda.xsd (xsi:noNamespaceSchemaLocation)

```xml
<ovoda xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="GIE0EJ_ovoda.xsd">
    <csoport nev="süni">
        <dolgozo id="1">
            <nev>Kiss Júlia</nev>
            <beosztas>óvónő</beosztas>
            <szulido>1989</szulido>
        </dolgozo>
        <dolgozo id="2">
            <nev>Nagy Mária</nev>
            <beosztas>dajka</beosztas>
            <szulido>1965</szulido>
        </dolgozo>
        <gyerek jel="alma">
            <nev>Szabó Péter</nev>
            <szulido>2020</szulido>
        </gyerek>
        <gyerek jel="körte">
            <nev>Okos Tóni</nev>
            <szulido>2019</szulido>
        </gyerek>
        <gyerek jel="virág">
            <nev>Kovács Márta</nev>
            <szulido>2019</szulido>
        </gyerek>
    </csoport>
        <csoport nev="maci">
        <dolgozo id="3">
            <nev>Gipsz Jolán</nev>
            <beosztas>óvónő</beosztas>
            <szulido>1980</szulido>
        </dolgozo>
        <dolgozo id="4">
            <nev>Szép Etelka</nev>
            <beosztas>dajka</beosztas>
            <szulido>1968</szulido>
        </dolgozo>
        <gyerek jel="alma">
            <nev>Kiss Tamás</nev>
            <szulido>2018</szulido>
        </gyerek>
        <gyerek jel="kocsi">
            <nev>Nagy Zoltán</nev>
            <szulido>2019</szulido>
        </gyerek>
        <gyerek jel="szilva">
            <nev>Takács Lili</nev>
            <szulido>2019</szulido>
        </gyerek>
    </csoport>

        <csoport nev="napocska">
        <dolgozo id="5">
            <nev>Varga Móni</nev>
            <beosztas>óvónő</beosztas>
            <szulido>1983</szulido>
        </dolgozo>
        <dolgozo id="6">
            <nev>Pásztor Erzsi</nev>
            <beosztas>dajka</beosztas>
            <szulido>1975</szulido>
        </dolgozo>
        <gyerek jel="dominó">
            <nev>Szabó Virág</nev>
            <szulido>2018</szulido>
        </gyerek>
        <gyerek jel="csillag">
            <nev>Teszt Elek</nev>
            <szulido>2019</szulido>
        </gyerek>
        <gyerek jel="alma">
            <nev>Horváth Sára</nev>
            <szulido>2019</szulido>
        </gyerek>
    </csoport>
</ovoda>
```

2. feladat

Az XML dokumentumhoz elkészítjük az XMLSchema-t.

```
http://www.w3.org/2001/XMLSchema (with embedded xml.xsd)
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="ovoda">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="csoport" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="dolgozo" maxOccurs="unbounded">
                            <xs:complexType>
                                 <xs:sequence>
                                    <xs:element name="nev" type="xs:string"/>
                                    <xs:element name="beosztas" type="xs:string"/>
                                    <xs:element name="szulido" type="xs:integer"/>
                                 </xs:sequence>
                                <xs:attribute type="xs:integer" name="id" use="required"/>
                            </xs:complexType>
                        </xs:element>
                        <xs:element name="gyerek" maxOccurs="unbounded">
                            <xs:complexType>
                                 <xs:sequence>
                                    <xs:element name="nev" type="xs:string"/>
                                    <xs:element name="szulido" type="xs:integer"/>
                                 </xs:sequence>
                                <xs:attribute type="xs:string" name="jel" use="required"/>
                            </xs:complexType>
                        </xs:element>
                    </xs:sequence>
                    <xs:attribute type="xs:string" name="nev" use="required"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:schema>
```

3. feladat

XQuery-vel eXistDB-ben lekérdezéseket készítünk.

Kilistázzuk az óvónőket abc sorrendben.

```
xquery version "3.1";

for $o in doc ("GIE0EJ_ovoda.xml") //dolgozo
where $o /beosztas = "óvónő"
order by $o /nev
return
    element {"óvónő"} {
        text {$o}
    }
```

A futtatás eredménye

```
1 <óvónő>
                        Gipsz Jolán
                        óvónő
                        1980
                </óvónő>
2 <óvónő>
                        Kiss Júlia
                        óvónő
                        1989
                </óvónő>
3 <óvónő>
                        Varga Móni
                        óvónő
                        1983
                </óvónő>
```

4. feladat

Kilistázzuk a „süni” csoportba járó gyerekek neveit.

```
1  xquery version "3.1";
2
3  for $cs in fn:doc ("GIE0EJ_ovoda.xml") //csoport
4  where $cs /@nev = "süni"
5  return
6      element {"sünicsoport"} {
7          text {$cs /gyerek/nev}
8      }
```

A futtatás eredménye

```
1 <sünicsoport>Szabó Péter Okos Tóni Kovács Mari Kocsis Laci</sünicsoport>
```

5. feladat

Lekérdezzük a legidősebb dajka nevét.

```
1  xquery version "3.1";
2
3  for $d in fn:doc ("GIE0EJ_ovoda.xml") //dolgozo
4  where $d /beosztas = "dajka" and $d /szulido = min (//szulido)
5  return
6      <legidosebbdajka>
7          {$d /nev}
8          {$d /szulido}
9      </legidosebbdajka>
```

A futtatás eredménye

```
1 <legidosebbdajka>
    <nev>Nagy Mária</nev>
    <szulido>1965</szulido>
</legidosebbdajka>
```

6. feladat

Új gyerek beszúrása a „süni” csoportba.

```
24          </gyerek>
25      <gyerek jel="fagyi">
26              <nev>Kocsis Laci</nev>
27              <szulido>2018</szulido>
28          </gyerek>
```

A futtatás eredménye

XML fájl:

```
21          <gyerek jel="virág">
22              <nev>Kovács Mari</nev>
23              <szulido>2019</szulido>
24          </gyerek>
25      <gyerek jel="fagyi">
26              <nev>Kocsis Laci</nev>
27              <szulido>2018</szulido>
28          </gyerek>
```

7. feladat

A Kovács Márta nevű gyermek nevének módosítása.

```
1  xquery version "3.1";
2
3  let $gy := fn:doc("GIE0EJ_ovoda.xml") //gyerek
4  return
5  update value //nev [.="Kovács Márta"] with "Kovács Mari"
```

A futtatás eredménye

```
21          <gyerek jel="virág">
22              <nev>Kovács Mari</nev>
23              <szulido>2019</szulido>
24          </gyerek>
```

8. feladat

A körte jelű gyerekek törlése.

```
1  xquery version "3.1";
2
3  let $gy := fn:doc("GIE0EJ_ovoda.xml") //gyerek[@jel = "körte"]
4
5  return
6      update delete $gy
```

9. feladat

Az alma jelű gyerekek kilistázása.

```
xquery version "3.1";

for $n in fn:doc("GIE0EJ_ovoda.xml")//gyerek
where $n/@jel="alma"
return
    element {"eredmény"} {
        attribute {"jel"} {$n/@jel},
        text {$n/nev}
    }
```

A futtatás eredménye

```
<eredmény jel="alma">Szabó Péter</eredmény>
<eredmény jel="alma">Kiss Tamás</eredmény>
<eredmény jel="alma">Horváth Sára</eredmény>
```

10. feladat

Életkor kiszámító függvény készítése.

```
xquery version "3.1";

declare namespace fv = "fl";
declare function fv:eletkor ($s) as xs:integer{
    let $e := fn:year-from-date(fn:current-date()) - $s cast as xs:integer
    return $e
};

<eredmeny>
{
    let $doc := fn:doc("GIE0EJ_ovoda.xml")
    for $gy in $doc //gyerek
    return
        element{"gyerek"} {
        attribute {"jel"} {$gy/@jel},
        attribute {"név"} {$gy/nev},
        text {fv:eletkor(number($gy/szulido))}
    }
}
</eredmeny>
```

A futtatás eredménye

```
<eredmeny>
    <gyerek jel="alma" név="Szabó Péter">3</gyerek>
    <gyerek jel="virág" név="Kovács Mari">4</gyerek>
    <gyerek jel="fagyi" név="Kocsis Laci">5</gyerek>
    <gyerek jel="alma" név="Kiss Tamás">5</gyerek>
    <gyerek jel="kocsi" név="Nagy Zoltán">4</gyerek>
    <gyerek jel="szilva" név="Takács Lili">4</gyerek>
    <gyerek jel="dominó" név="Szabó Virág">5</gyerek>
    <gyerek jel="csillag" név="Teszt Elek">4</gyerek>
    <gyerek jel="alma" név="Horváth Sára">4</gyerek>
</eredmeny>
```

11. feladat
Az egyes csoportok létszámának meghatározása.

```
xquery version "3.1";

for $cs in fn:doc("GIE0EJ_ovoda.xml")//csoport
return
    element {"létszám"} {
        attribute {"csoport"} {$cs/@nev},
        text {count($cs/gyerek)}
    }
```

A futtatás eredménye

```
<létszám csoport="süni">3</létszám>

<létszám csoport="maci">3</létszám>

<létszám csoport="napocska">3</létszám>
```

MyBatis

A feladat leírása
ORM programozás és MyBatis SQL-lel.
Téma: Könyvek
Fejlesztő környezet: SQL Developer, Eclipse

A feladat elkészítésének lépései
1. feladat
Adatbázis elkészítése.

```
CREATE TABLE Books(
isbn VARCHAR2(14) NOT NULL,
title VARCHAR2(50),
price NUMBER(10),
PRIMARY KEY(isbn)
)
```

2. feladat
XML config fájl létrehozása.

```
-//mybatis.org//DTD Config 3.0//EN (doctype)
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE configuration
    PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
    "http://mybatis.org/dtd/mybatis-3-config.dtd">
<configuration>
    <typeAliases>
        <typeAlias type="main.mybatis.Book" alias="book" />
    </typeAliases>
    <environments default="development">
        <environment id="development">
            <transactionManager type="JDBC"/>
            <dataSource type="POOLED">
                <property name="driver" value="oracle.jdbc.OracleDriver"/>
                <property name="url" value="jdbc:oracle:thin:@localhost:1521:XE"/>
                <property name="username" value="system" />
                <property name="password" value="orql" />
            </dataSource>
        </environment>
    </environments>
    <mappers>
        <mapper resource="mybatis/BookMapper.xml"/>
    </mappers>
</configuration>
```

3. feladat

Mapping fájl létrehozása.

```
-//mybatis.org//DTD Mapper 3.0//EN (doctype)
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper
    PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
    "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace = "resources.mybatis.BookMapper">
    <resultMap id="bookResult" type="book">
        <id property = "isbn" column="isbn"/>
        <result property = "title" column = "title" />
        <result property = "price" column = "price" />
    </resultMap>

    <select id = "findAllBooks" resultType = "book" resultMap = "bookResult">
        SELECT isbn, title, price from Books
    </select>

    <insert id="insertBook" parameterType = "book" keyProperty="isbn">
        INSERT into Books VALUES(#{isbn},#{title},#{price})
    </insert>

    <update id="updateBook" parameterType="book">
        UPDATE Books SET price = #{price} WHERE isbn = #{isbn}
    </update>

    <delete id="deleteBook" parameterType="string">
        DELETE FROM Books WHERE isbn = #{isbn}
    </delete>

    <select id="selectBook" resultType="book" resultMap="bookResult">
        SELECT isbn, title, price FROM Books WHERE price > 5000
    </select>
</mapper>
```

4. feladat

Book class létrehozása.

```java
package main.mybatis;

public class Book {
    private String isbn;
    private String title;
    private int price;

    public String getIsbn() {
        return isbn;
    }

    public void setIsbn(String isbn) {
        this.isbn = isbn;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public int getPrice() {
        return price;
    }

    public void setPrice(int price) {
        this.price = price;
    }
}
```

5. feladat

Session manager class létrehozása.

```java
package main.mybatis;

import java.io.InputStream;

public class MyBatisUtil{
    private static SqlSessionFactory sqlSessionFactory;
    static {
        String resource = "mybatis-config.xml";
        InputStream inputStream;
        try {
            inputStream = Resources.getResourceAsStream(resource);
            sqlSessionFactory = new SqlSessionFactoryBuilder().build(inputStream);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static SqlSessionFactory getSqlSessionFactory() {
        return sqlSessionFactory;
    }

}
```

6. feladat

DAO class létrehozása.

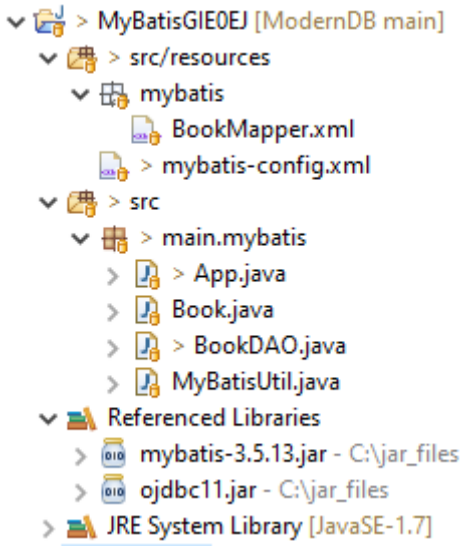
```java
package main.mybatis;
import java.util.List;

public class BookDAO {

    public void insert(Book book) {
        SqlSession session = MyBatisUtil.getSqlSessionFactory().openSession();
        session.insert("resources.mybatis.BookMapper.insertBook", book);
        session.commit();
        session.close();
    }

    public void update(Book book) {
        SqlSession session = MyBatisUtil.getSqlSessionFactory().openSession();
        session.update("resources.mybatis.BookMapper.updateBook", book);
        session.commit();
        session.close();
    }

    public List<Book> findAll() {
        SqlSession session = MyBatisUtil.getSqlSessionFactory().openSession();
        List<Book> books = session.selectList("resources.mybatis.BookMapper.findAllBooks");
        session.commit();
        session.close();
        return books;
    }

    public List<Book> select() {
        SqlSession session = MyBatisUtil.getSqlSessionFactory().openSession();
        List<Book> books = session.selectList("resources.mybatis.BookMapper.selectBook");
        session.commit();
        session.close();
        return books;
    }

    public void delete(String id) {
        SqlSession session = MyBatisUtil.getSqlSessionFactory().openSession();
        session.delete("resources.mybatis.BookMapper.deleteBook", id);
        session.commit();
        session.close();
    }
}
```

7. feladat

Application class létrehozása.

```java
package main.mybatis;

import java.util.List;

public class App {
    public static void main(String[] args) {
        BookDAO bookDAO = new BookDAO();

        //INSERT
        Book bookIn = new Book();
        bookIn.setIsbn("006");
        bookIn.setTitle("Aranyember");
        bookIn.setPrice(2500);
        bookDAO.insert(bookIn);
        System.out.println("Mentve: isbn: " + bookIn.getIsbn() + ", title: " + bookIn.getTitle() + ", price: " + bookIn.getPrice() + "\n");

        //UPDATE
        Book bookUp = new Book();
        bookUp.setIsbn("002");
        bookUp.setPrice(8200);
        bookDAO.update(bookUp);
        System.out.println("Módosítva: isbn: " + bookUp.getIsbn() + ", title: " + bookUp.getTitle() + ", price: " + bookUp.getPrice() + "\n");

        //FINDALL
        System.out.println("Összes könyv: ");
        List<Book> allBooks = bookDAO.findAll();
        for(Book b : allBooks) {
            System.out.println("isbn: " + b.getIsbn() + ", title: " + b.getTitle() + ", price: " + b.getPrice() + "\n");
        }

        //SELECT
        List<Book> listedBooks = bookDAO.select();
        System.out.println("5000-nél drágább könyvek: ");
        for (Book b : listedBooks) {
            System.out.println("isbn: " + b.getIsbn() + ", title: " + b.getTitle() + ", price: " + b.getPrice() + "\n");
        }

        //DELETE
        bookDAO.delete("006");
        System.out.println("Rekord törölve\n");
    }
}
```

A könyvtár szerkezete a következő:



A futtatás eredménye

SQL futtatás eredménye

	ISBN	TITLE	PRICE
1	001	Egri Csillagok	4000
2	002	Pál utcai fiúk	3200
3	003	Robin Hood	5200
4	004	Anyegin	3800
5	005	Az ajtó	4300

A Java projekt futtatásának eredménye

```
Mentve: isbn: 006, title: Aranyember, price: 2500

Módosítva: isbn: 002, price: 8200

Összes könyv:
isbn: 001, title: Egri Csillagok, price: 4000

isbn: 002, title: Pál utcai fiúk, price: 8200

isbn: 003, title: Robin Hood, price: 5200

isbn: 004, title: Anyegin, price: 3800

isbn: 005, title: Az ajtó, price: 4300

isbn: 006, title: Aranyember, price: 2500

5000-nél drágább könyvek:
isbn: 002, title: Pál utcai fiúk, price: 8200

isbn: 003, title: Robin Hood, price: 5200

Rekord törölve
```

MongoDB

A feladat leírása

MongoDB használata, programozása, Maven projekt
Téma: Autók és tulajdonosai
Fejlesztő környezet: Eclipse, MongoDB, Robo 3T

1. feladat

Az adatbázis és kollekció létrehozása és feltöltése adatokkal.

```
use MDB_GIE0EJ
```

0 sec.

```
switched to db MDB_GIE0EJ
```

```
db.createCollection("auto")
```

0.017 sec.

```
db.auto.insertMany([{"tipus":"Suzuki","szin":"sarga","ar":600000,"gyev":2005},
{"tipus":"Toyota","szin":"feher","ar":1600000,"gyev":2010},
{"tipus":"Wolkswagen","szin":"zold","ar":904000,"gyev":2007},
{"tipus":"Audi","szin":"kek","ar":3220000,"gyev":2014},
{"tipus":"Opel","szin":"piros","ar":500011,"gyev":2001}])
```

0.002 sec.

Számoljuk meg azokat az autókat, amelyeknek az ára több mint 1.000.000.

```
db.auto.find({"ar":{"$gt":1000000}}).count()
```

0.001 sec.

```
2
```

Adjunk a rekordokhoz egy új mezőt „állapot” néven és állítsuk „jó”-ra.

```
db.auto.updateMany({},{"$set":{"allapot":"jo"}})
```

0.007 sec.

A 2004 előtt gyártott autók állapotát állítsuk sérültre.

```
db.auto.updateMany({"gyev":{"$lt":2004}},{"$set":{"allapot":"serult"}})
```

0.002 sec.

1. feladat eredménye

Key	Value	Type
(1) ObjectId("647573313000572...	{ 6 fields }	Object
_id	ObjectId("647573313000572bc2f1d9a6")	ObjectId
tipus	Suzuki	String
szin	sarga	String
ar	600000.0	Double
gyev	2005.0	Double
allapot	jo	String
(2) ObjectId("647573313000572...	{ 6 fields }	Object
_id	ObjectId("647573313000572bc2f1d9a7")	ObjectId
tipus	Toyota	String
szin	feher	String
ar	1600000.0	Double
gyev	2010.0	Double
allapot	jo	String
(3) ObjectId("647573313000572...	{ 6 fields }	Object
_id	ObjectId("647573313000572bc2f1d9a8")	ObjectId
tipus	Wolkswagen	String
szin	zold	String
ar	904000.0	Double
gyev	2007.0	Double
allapot	jo	String
(4) ObjectId("647573313000572...	{ 6 fields }	Object
_id	ObjectId("647573313000572bc2f1d9a9")	ObjectId
tipus	Audi	String
szin	kek	String
ar	3220000.0	Double
gyev	2014.0	Double
allapot	jo	String
(5) ObjectId("647573313000572...	{ 6 fields }	Object
_id	ObjectId("647573313000572bc2f1d9aa")	ObjectId
tipus	Opel	String
szin	piros	String
ar	500011.0	Double
gyev	2001.0	Double
allapot	serult	String

2. feladat

Készítsünk egy tulajdonos kollekciót, amely név és kor mezőkkel rendelkezik.

```
db.createCollection("tulajdonos")
```

0.011 sec.

Készítsünk egy tárolt függvényt, amely az ID, név és kor megadásával új tulajdonosokat tud felvenni tulajdonos kollekcióba.

```
db.system.js.save({
  _id: "save_tulaj",
  value: function (id, nev, kor) {
    db.tulajdonos.insertOne({
      _id: id,
      nev: nev,
      kor: kor
    });
  }
})
```

0.002 sec.

```
Updated 1 new record(s) in 2ms
```

Készítsünk tárolt függvényt, amellyel új rekordot tudunk felvenni az auto kollekcióba, melyhez hozzáadunk egy tulaj mezőt is, mely a tulajdonos tábla elemeinek az ID-jét tartalmazza.

```
db.system.js.save({
  _id: "save_auto",
  value: function (tipus, szin, ar, gyev, allapot, tulaj) {
    var tulajdonos = db.tulajdonos.findOne({_id: tulaj});
    if (tulajdonos) {
      db.auto.insertOne({
        tipus: tipus,
        szin: szin,
        ar: ar,
        gyev: gyev,
        allapot: allapot,
        tulaj: tulajdonos._id
      });
    } else {
      print("Hiba: A megadott tulajdonos nem található!");
    }
  }
});
```

0.002 sec.

```
Updated 1 new record(s) in 1ms
```

Készítsünk tárolt függvényt, amely a paraméterként kapott tulajdonos neve alapján kilistázza az összes olyan nevű tulajdonost!

```
db.system.js.save({
  _id: "getTulajByName",
  value: function (nev) {
    return db.tulajdonos.find({ nev: nev }).toArray();
  }
});
```

0.003 sec.

```
Updated 1 new record(s) in 3ms
```

Készítsünk tárolt függvényt, amely a paraméterként kapott tulajdonos neve alapján kilistázza az összes hozzá tartozó autót!

```
db.system.js.save({
  _id: "getAutoByTulajName",
  value: function (tulajnev) {
    var tulajdonos = db.tulajdonos.findOne({nev: tulajnev});
    if (tulajdonos) {
      return db.auto.find({tulaj: tulajdonos._id}).toArray();
    } else {
      print("Nem található tulajdonos a megadott névvel!");
      return [];
    }
  }
});
```

0.002 sec.

```
Updated 1 new record(s) in 2ms
```

Csökkentsük a sérült állapotú autók árát 300.000 – el!

```
db.auto.updateMany(
  { allapot: "sérült" },
  { $inc: { ar: -300000 } }
)
```

0.006 sec.

A $where használatával számoljuk meg azokat az autókat, amelyeknek az ára kisebb, mint 1.000.000. és a gyártás éve 2010 előtti!

```
db.auto.find({
  $where: function() {
    return this.ar < 1000000 && this.gyev < 2010;
  }
}).count();
```

0.111 sec.

```
3
```

Írassuk ki típus szerint csoportosítva az átlagárat.

```
db.auto.aggregate([
  {
    $group: {
      _id: "$tipus",
      atlagar: { $avg: "$ar" }
    }
  }
]);
```

auto 0.005 sec.

Key	Value
(1) Wolkswagen	{ 2 fields }
_id	Wolkswagen
atlagar	904000.0
(2) Suzuki	{ 2 fields }
_id	Suzuki
atlagar	600000.0
(3) Audi	{ 2 fields }
_id	Audi
atlagar	3220000.0
(4) Toyota	{ 2 fields }
_id	Toyota
atlagar	1600000.0
(5) Opel	{ 2 fields }
_id	Opel
atlagar	1750005.5

Írassuk ki csökkenő sorrendben, hogy típusonként mennyi autó rendelkezik sérült státusszal.

```
db.auto.aggregate([
  {
    $match: {
      allapot: "sérült"
    }
  },
  {
    $group: {
      _id: "$tipus",
      count: { $sum: 1 }
    }
  },
  {
    $sort: {
      count: -1
    }
  }
]);
```

auto 0.001 sec.

Key	Value	Type
(1) BMW	{ 2 fields }	Object
_id	BMW	String
count	2.0	Double
(2) Opel	{ 2 fields }	Object
_id	Opel	String
count	2.0	Double
(3) Toyota	{ 2 fields }	Object
_id	Toyota	String
count	1.0	Double

2. feladat futtatásai

save_tulaj függvény meghívása:

```
db.loadServerScripts();
save_tulaj("T5", "Janka", 20);
```

Script executed successfully, but there are no results to show.

```
db.getCollection('tulajdonos').find({})
```

tulajdonos 0.002 sec.

Key	Value
(1) T5	{ 3 fields }
_id	T5
nev	Janka
kor	20.0

save_auto függvény meghívása:

```
db.loadServerScripts();
save_auto("Opel","fehér",3000000,2010, "jó", "T5");
```

Script executed successfully, but there are no results to show.

Key	Value	Type
(1) ObjectId("647573313000572...	{ 6 fields }	Object
(2) ObjectId("647573313000572...	{ 6 fields }	Object
(3) ObjectId("647573313000572...	{ 6 fields }	Object
(4) ObjectId("647573313000572...	{ 6 fields }	Object
(5) ObjectId("647573313000572...	{ 6 fields }	Object
_id	ObjectId("647573313000572bc2f1d9aa")	ObjectId
tipus	Opel	String
szin	piros	String
ar	500011.0	Double
gyev	2001.0	Double
allapot	serult	String
(6) ObjectId("647577903000572...	{ 7 fields }	Object
_id	ObjectId("647577903000572bc2f1d9ab")	ObjectId
tipus	Opel	String
szin	fehér	String
ar	3000000.0	Double
gyev	2010.0	Double
allapot	jó	String
tulaj	T5	String

getTulajByName függvény meghívása:

```
db.loadServerScripts();
getTulajByName("Janka");
```

0.001 sec.

Key	Value	Type
(1)	[1 element]	Array
[0]	{ 3 fields }	Object
_id	T5	String
nev	Janka	String
kor	20.0	Double

getAutoByTulajName függvény futtatása:

```
db.loadServerScripts();
getAutoByTulajName("Janka");
```

0.002 sec.

Key	Value	Type
(1)	[1 element]	Array
[0]	{ 7 fields }	Object
_id	ObjectId("647577903000572bc2f1d9ab")	ObjectId
tipus	Opel	String
szin	fehér	String
ar	3000000.0	Double
gyev	2010.0	Double
allapot	jó	String
tulaj	T5	String

3. feladat

Létrehozunk egy Maven projektet, amelyben a pom.xml fájlba először beillesszük a függőséget.

```
<dependencies>
  <dependency>
    <groupId>org.mongodb</groupId>
    <artifactId>mongo-java-driver</artifactId>
    <version>3.12.10</version>
  </dependency>
</dependencies>
```

Csatlakozunk az adatbázishoz.

```
package MDB_gyak.MDB_GIE0EJ;

import org.bson.Document;

public class GIE0EJ
{
    public static void main( String[] args )
    {
        MongoClient mongoClient = MongoClients.create("mongodb://localhost:27017");
        MongoDatabase db = mongoClient.getDatabase("MDB_GIE0EJ");
        MongoCollection<Document> auto = db.getCollection("auto");
```

Írjunk egy metódust, amely kilistázza az auto kollekció rekordjait.

```
71    public static void listAll(MongoCollection<Document> auto)
72    {
73        FindIterable<Document> iterDoc = auto.find();
74        int i = 1;
75        Iterator<Document> it = iterDoc.iterator();
76        while (it.hasNext()) {
77            System.out.println(it.next());
78            i++;
79        }
80    }
81 }
```

Írjunk egy metódust, amely új elemet vesz fel az auto kollekcióba.

```
61    public static void insertOne(MongoCollection<Document> auto) {
62        Document doc = new Document("tipus", "javaInserted")
63                .append("szin", "kék")
64                .append("ar", 2000110)
65                .append("gyev", 2009)
66                .append("allapot", "jó");
67
68        auto.insertOne(doc);
69    }
```

Írjunk olyan metódust, mely egyszerre több elemet vesz fel a kollekcióba.

```
44    public static void insertMany(MongoCollection<Document> auto) {
45        List<Document> autoList = Arrays.asList(
46                new Document().append("tipus", "javaInserted")
47                .append("szin", "piros")
48                .append("ar", 3000500)
49                .append("gyev", 2010)
50                .append("allapot", "jó"),
51
52                new Document().append("tipus", "javaInserted")
53                .append("szin", "zöld")
54                .append("ar", 1300200)
55                .append("gyev", 2008)
56                .append("allapot", "jó"));
57
58        auto.insertMany(autoList);
59    }
```

Írjunk metódust, amely sérült állapotúra módosítja az adott típusú autókat.

```
38    public static void update(MongoCollection<Document> auto) {
39        Document tipus = new Document("tipus", "javaInserted");
40        Document allapot = new Document("$set", new Document("allapot", "sérült"));
41        auto.updateMany(tipus, allapot);
42    }
```

Írjunk metódust, amely törli az adott típusú rekordokat.

```java
    public static void delete(MongoCollection<Document> auto) {
        Document tipus = new Document("tipus", "javaInserted");
        auto.deleteMany(tipus);
    }
```

3. feladat futtatása

```
Updated collection
Document{{_id=647573313000572bc2f1d9a6, tipus=Suzuki, szin=sarga, ar=600000.0, gyev=2005.0, allapot=jo}}
Document{{_id=647573313000572bc2f1d9a7, tipus=Toyota, szin=feher, ar=1600000.0, gyev=2010.0, allapot=jo}}
Document{{_id=647573313000572bc2f1d9a8, tipus=Wolkswagen, szin=zold, ar=904000.0, gyev=2007.0, allapot=jo}}
Document{{_id=647573313000572bc2f1d9a9, tipus=Audi, szin=kek, ar=3220000.0, gyev=2014.0, allapot=jo}}
Document{{_id=647573313000572bc2f1d9aa, tipus=Opel, szin=piros, ar=500011.0, gyev=2001.0, allapot=serult}}
Document{{_id=647577903000572bc2f1d9ab, tipus=Opel, szin=fehér, ar=3000000.0, gyev=2010.0, allapot=jó, tulaj=T5}}
Document{{_id=64757bfc3000572bc2f1d9ac, tipus=BMW, szin=zöld, ar=1.2E7, gyev=2008.0, allapot=sérült}}
Document{{_id=64757bfc3000572bc2f1d9ad, tipus=Audi, szin=fekete, ar=4001100.0, gyev=2007.0, allapot=jó}}
Document{{_id=64757bfc3000572bc2f1d9ae, tipus=Opel, szin=zöld, ar=1.2E7, gyev=2008.0, allapot=sérült}}
Document{{_id=64757bfc3000572bc2f1d9af, tipus=Toyota, szin=fehér, ar=4221100.0, gyev=2000.0, allapot=sérült}}
Document{{_id=64757bfc3000572bc2f1d9b0, tipus=BMW, szin=zöld, ar=6001100.0, gyev=2008.0, allapot=jó}}
Document{{_id=64757c333000572bc2f1d9b1, tipus=BMW, szin=zöld, ar=1.2E7, gyev=2008.0, allapot=sérült}}
Document{{_id=64757c333000572bc2f1d9b2, tipus=Audi, szin=fekete, ar=4001100.0, gyev=2007.0, allapot=jó}}
Document{{_id=64757c333000572bc2f1d9b3, tipus=Opel, szin=zöld, ar=1.2E7, gyev=2008.0, allapot=sérült}}
Document{{_id=64757c333000572bc2f1d9b4, tipus=BMW, szin=zöld, ar=6001100.0, gyev=2008.0, allapot=jó}}
Document{{_id=647580dde07c067d73231202, tipus=javaInserted, szin=kék, ar=2000110, gyev=2009, allapot=sérült}}
Document{{_id=647580dde07c067d73231203, tipus=javaInserted, szin=piros, ar=3000500, gyev=2010, allapot=sérült}}
Document{{_id=647580dde07c067d73231204, tipus=javaInserted, szin=zöld, ar=1300200, gyev=2008, allapot=sérült}}
```