

Jegyzőkönyv

Adatkezelés XML környezetben

Féléves feladat

Készítette: Dékány Péter

Neptunkód: FHI2ET

A feladat leírása:

A feladatom a borászatok és az azok által készített borok nyilvántartása. A borok mellett a borászatok tulajdonosai és a vevők.

.

Az egyedek:

- Borászat,
- Tulajdonos (borászat tulajdonosa),
- Bor,
- Vevő,

Ezen egyedek tulajdonságai:

- Borászat:
 - o Btkód: A Borászat elsődleges kulcsa
 - o Név: A Borászat neve
 - o Cím: Város, Irszám, Utca, Hszám
 - o Alap: A Borászat alapítása
- Tulajdonos:
 - o Tkód: A Tulajdonos elsődleges kulcsa
 - o Név: A Tulajdonoshoz tartozó név
 - o Szid: A Tulajdonoshoz tartozó születési idő
 - o Szh: A Tulajdonoshoz tartozó születési hely
- Bor:
 - o Bkód: A Bor elsődleges kulcsa
 - o Név: A Bor neve
 - o Év: A Szüretelés éve
 - o Fajta: A szőlő fajtája
 - o Ár: A Bor ára

- Vevő

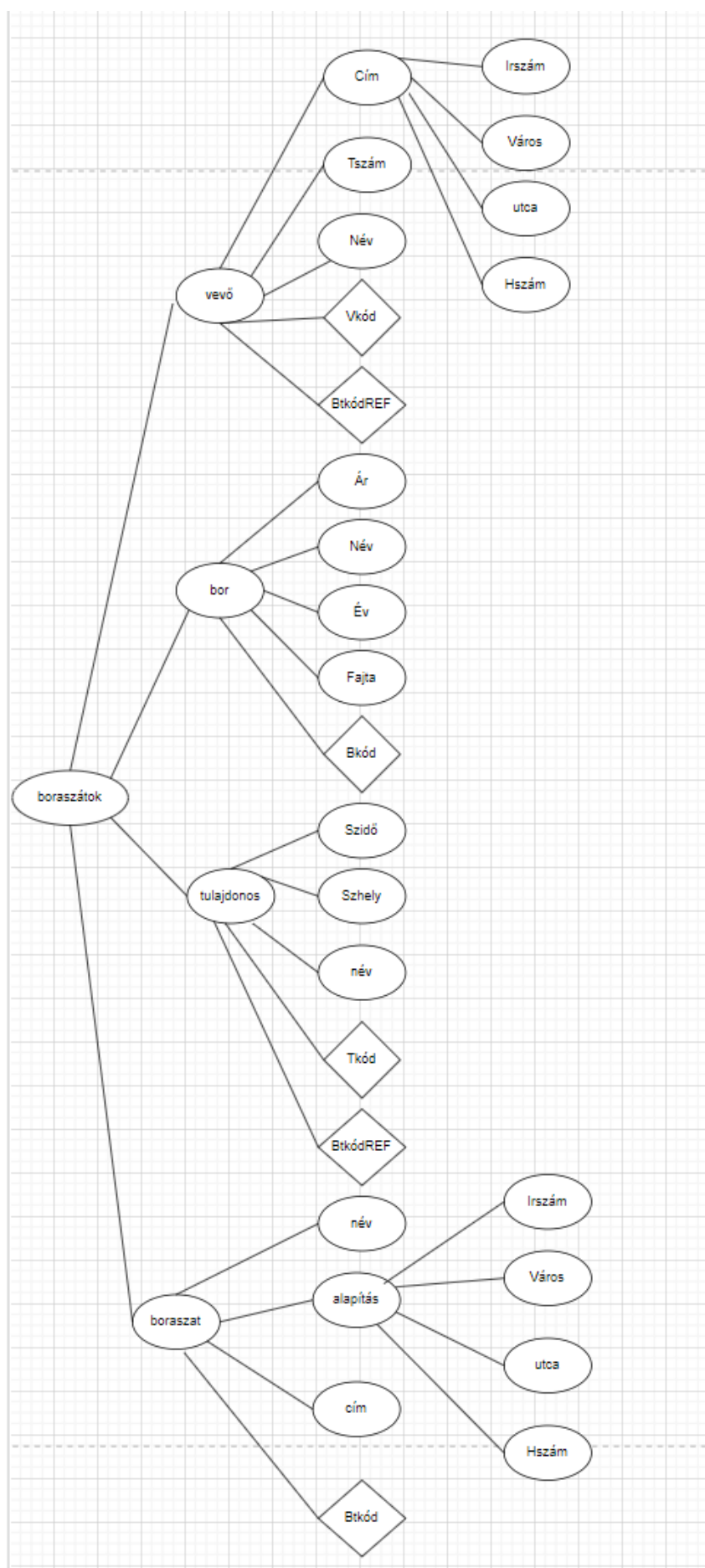
- o V kód: A Vevő elsődleges kulcsa
- o Név: A Vevő neve
- o Cím: Város, Irszám, Utca, Hszám
- o Tszám: A Vevő telefonszáma

1:1 kapcsolat: Tulajdonos - Születési idő

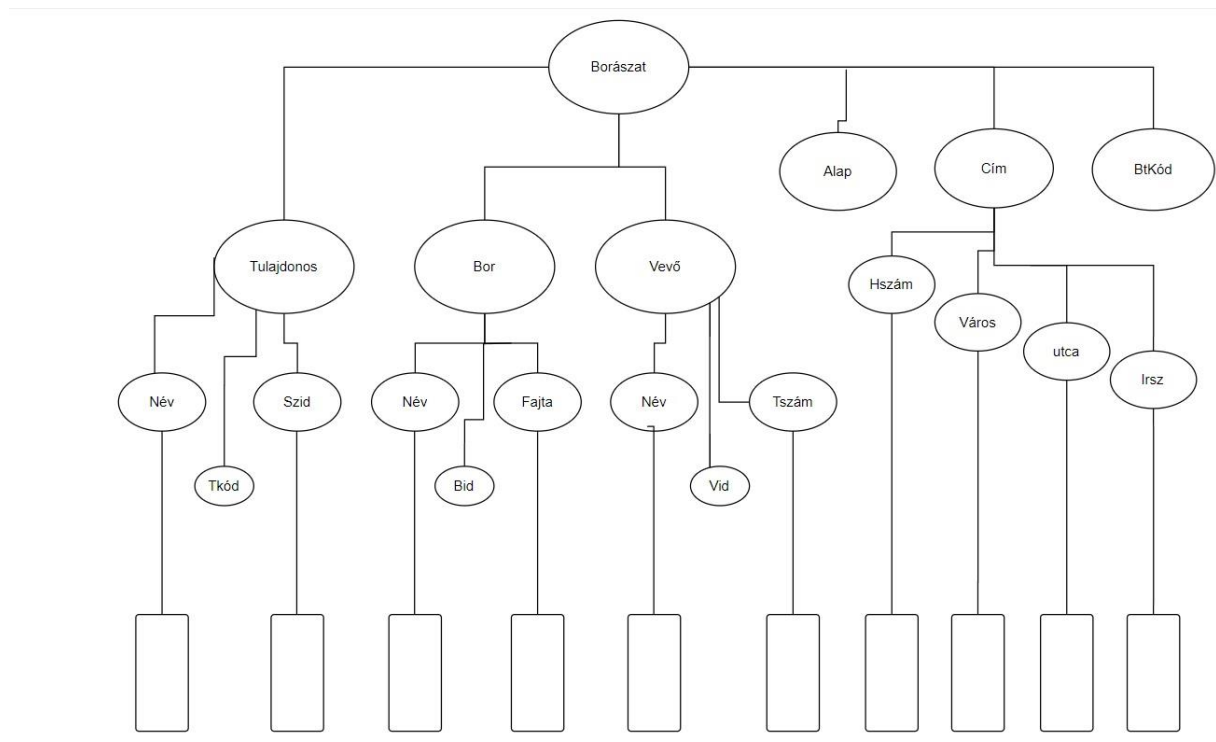
1:N kapcsolat: Tulajdonos – Borászatok

N:M kapcsolat: Tulajdonos - Borok

1 a) ER modell



1 b) XDM modell

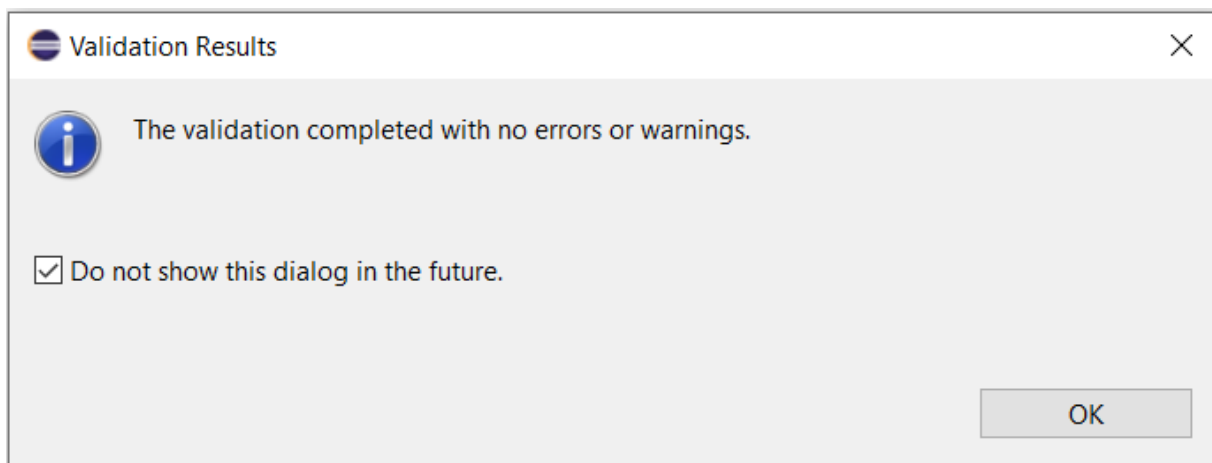


1 c) Az XDM modell alapján XML dokumentum készítése:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-model href="FHI2ET.xsd" type="application/xml"
schematypens="http://www.w3.org/2001/XMLSchema"?>
```

```
<boraszatok>
  <boraszat Btkod="001">
    <nev>Laboda</nev>
    <alap>1997</alap>
    <cim>
      <varos>Miskolc</varos>
      <Irsz>3515</Irsz>
      <Hszam>14</Hszam>
      <utca>Salétrom</utca>
    </cim>
  </boraszat>
  <boraszat Btkod="002">
    <nev>Uri</nev>
    <alap>1996</alap>
    <cim>
      <varos>Csemő</varos>
      <Irsz>3745</Irsz>
      <Hszam>1</Hszam>
      <utca>Vass</utca>
    </cim>
  </boraszat>
</boraszatok>
```

```
        </cim>
    </boraszat>
    <bor Bkod="010" BtkodREF="001">
        <nev>Kóbor</nev>
        <ar>1000</ar>
        <ev>2001</ev>
        <fajta>Tablettás</fajta>
    </bor>
    <bor Bkod="020" BtkodREF="002">>
        <nev>Dorombor</nev>
        <ar>2300</ar>
        <ev>2021</ev>
        <fajta>Irsai</fajta>
    </bor>
    <tulajdonos Tkod="100" BtkodREF="001">>
        <nev>Laboda Dániel</nev>
        <Szid>1970</Szid>
        <Szh>Miskolc</Szh>
    </tulajdonos>
    <tulajdonos Tkod="200" BtkodREF="002">>
        <nev>Uri-Kovács Bence</nev>
        <Szid>1964</Szid>
        <Szh>Csemő</Szh>
    </tulajdonos>
    <vevo Vkod="111">
        <nev>Róthi János</nev>
        <tszam>06209365850</tszam>
        <cim>
            <irszam>3650</irszam>
            <varos>Kurityán</varos>
            <utca>Szabad</utca>
            <hszam>11</hszam>
        </cim>
    </vevo>
    <vevo Vkod="222">
        <nev>sirnakó Boldizsár</nev>
        <tszam>06202345850</tszam>
        <cim>
            <irszam>4250</irszam>
            <varos>Szabadka</varos>
            <utca>Foglaltka</utca>
            <hszam>41</hszam>
        </cim>
    </vevo>
</boraszatok>
```



1 d) Az XML dokumentum alapján az XMLSchema készítése:

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

```
<!-- Egyszerű elemek-->
```

```
<xs:element name="nev" type="xs:string"/>
<xs:element name="alap" type="xs:integer"/>
<xs:element name="varos" type="xs:string"/>
<xs:element name="Irsz" type="xs:integer"/>
<xs:element name="utca" type="xs:string"/>
<xs:element name="Hszam" type="xs:integer"/>
<xs:element name="ar" type="xs:integer"/>
<xs:element name="ev" type="xs:integer"/>
<xs:element name="fajta" type="xs:string"/>
<xs:element name="Szid" type="xs:integer"/>
<xs:element name="Szh" type="xs:string"/>
<xs:element name="tszam" type="xs:string"/>
<xs:element name="nev" type="xs:string"/>
```

```
<!--Attribútumok-->
```

```
<xs:attribute name="Btkod" type="xs:integer"/>
<xs:attribute name="Bkod" type="xs:integer"/>
<xs:attribute name="Tkod" type="xs:integer"/>
<xs:attribute name="Vkod" type="xs:integer"/>
<xs:attribute name="BtkodREF" type="xs:integer"/>
```

```
<!--Komplex elemek: alapok-->
```

```
<xs:complexType name="cimtype">
  <xs:sequence>
    <xs:element ref="Varos">
```

```

        <xs:element ref="Irsz">
        <xs:element ref="utca">
        <xs:element ref="Hszam">
    </xs:sequence>
</xs:complexType>

<xs:complexType name="boraszattype">
    <xs:sequence>
        <xs:element ref="nev">
        <xs:element ref="alap">
        <xs:element type="cimtype" name="cim">
    </xs:sequence>
    <xs:attribute ref="Btkod" use="required"/>
</xs:complexType>

<xs:complexType name="bortype">
    <xs:sequence>
        <xs:element ref="nev">
        <xs:element ref="ar">
        <xs:element ref="fajta">
        <xs:element ref="ev">
    </xs:sequence>
    <xs:attribute ref="Bkod" use="required"/>
    <xs:attribute ref="BtkodREF" use="required"/>
</xs:complexType>

<xs:complexType name="tulajdonostype">
    <xs:sequence>
        <xs:element ref="nev">
        <xs:element ref="Szid">
        <xs:element ref="Szh">
        <xs:element type="cimtype" name="cim">
    </xs:sequence>
    <xs:attribute ref="Tkod" use="required"/>
    <xs:attribute ref="BtkodREF" use="required"/>
</xs:complexType>

<xs:complexType name="vevotype">
    <xs:sequence>
        <xs:element ref="nev">
        <xs:element ref="tszam">
        <xs:element type="cimtype" name="cim">
    </xs:sequence>
    <xs:attribute ref="Vkod" use="required"/>
</xs:complexType>

<!-- Kész s ma -->

```



```

<xs:element name="boraszatok">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="boraszat" type="boraszattype"/>
      <xs:element name="bor" type="bortype"/>
      <xs:element name="tulajdonos" type="tulajdonostype"/>
      <xs:element name="vevo" type="vevotype"/>
    </xs:sequence>
  </xs:complexType>

  <xs:key name="Btkod">
    <xs:selector xpath="boraszatok/boraszat"/>
    <xs:field xpath="@Btkod"/>
  </xs:key>
  <xs:key name="Bkod">
    <xs:selector xpath="boraszatok/bor"/>
    <xs:field xpath="@Bkod"/>
  </xs:key>
  <xs:key name="Tkod">
    <xs:selector xpath="boraszatok/tulajdonos"/>
    <xs:field xpath="@Tkod"/>
  </xs:key>
  <xs:key name="Vkod">
    <xs:selector xpath="boraszatok/Vevo"/>
    <xs:field xpath="@Vkod"/>
  </xs:key>
  <xs:keyref name="BtkodREF" refer="Btkod">
    <xs:selector xpath="boraszatok/bor"/>
    <xs:field xpath="@BtkodREF"/>
  </xs:keyref>
  <xs:keyref name="BtkodREF" refer="Btkod">
    <xs:selector xpath="boraszatok/tulajdonos"/>
    <xs:field xpath="@BtkodREF"/>
  </xs:keyref>
</xs:element>
</xs:schema>

```

2 a) Adatolvasás

```

package DOMreadFHI2ET;

import java.io.File;
import java.io.IOException;
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 DOMreadFHI2ET {
    public static void main(String[] args) {
        try{
            File xmlDoc = new File("src/DOMreadFHI2ET/FHI2ET.xml");
            DocumentBuilderFactory dbFact =
DocumentBuilderFactory.newInstance();
            DocumentBuilder dBuild = dbFact.newDocumentBuilder();
            Document doc = dBuild.parse(xmlDoc);

            //gyökér elem
            System.out.println("Gyökér elem: "+
doc.getDocumentElement().getNodeName());
            System.out.println("\n");

            //boraszat beolvasasa
            NodeList nList = doc.getElementsByTagName("boraszat");
            for(int i=0; i<nList.getLength();i++)
            {
                Node nNode = nList.item(i);

                if(nNode.getNodeType() == Node.ELEMENT_NODE)
                {
                    Element eElement = (Element) nNode;
                    //boraszat adatai
                    System.out.println("boraszat Id: " +
eElement.getAttribute("Btkod"));
                    System.out.println("Boraszat neve: " +

eElement.getElementsByTagName("nev").item(0).getTextContent());
                    System.out.println("Boraszat alapitasa: " +

eElement.getElementsByTagName("alap").item(0).getTextContent());

                    System.out.println("Boraszat cime: " + "\n"+

eElement.getElementsByTagName("varos").item(0).getTextContent()+
"+eElement.getElementsByTagName("Irsz").item(0).getTextContent()
                    +",
"+eElement.getElementsByTagName("utca").item(0).getTextContent()+" "+

eElement.getElementsByTagName("Hszam").item(0).getTextContent()+". " );

                }
            }
            System.out.println("_____");

            NodeList nList11 = doc.getElementsByTagName("tulajdonos");
            for(int i=0; i<nList11.getLength();i++)
            {
                Node nNode = nList11.item(i);

                if(nNode.getNodeType() == Node.ELEMENT_NODE)
                {
                    Element eElement = (Element) nNode;
                    //tulajdonos adatai

```

```

        System.out.println("tulajdonos Id: " +
eElement.getAttribute("Tkod"));
        System.out.println("Tulajdonos neve: " +
        eElement.getElementsByTagName("nev").item(0).getTextContent());
        System.out.println("Tulajdonos születési ideje: "
+
        eElement.getElementsByTagName("Szid").item(0).getTextContent());
        System.out.println("Tulajdonos születési helye: "
+
        eElement.getElementsByTagName("Szh").item(0).getTextContent());
    }
}

System.out.println("_____");

NodeList nList1 = doc.getElementsByTagName("bor");
for(int i=0; i<nList1.getLength();i++)
{
    Node nNode = nList1.item(i);

    if(nNode.getNodeType() == Node.ELEMENT_NODE)
    {
        Element eElement = (Element) nNode;
        //Bor adatai
        System.out.println("bor Id: " +
eElement.getAttribute("Bkod"));
        System.out.println("Bor neve: " +
        eElement.getElementsByTagName("nev").item(0).getTextContent());
        System.out.println("Bor ara: " +
        eElement.getElementsByTagName("ar").item(0).getTextContent());
        System.out.println("Bor fajta: " +
        eElement.getElementsByTagName("fajta").item(0).getTextContent());
    }
}
System.out.println("_____");

NodeList nList111 = doc.getElementsByTagName("vevo");
for(int i=0; i<nList111.getLength();i++)
{
    Node nNode = nList111.item(i);

    if(nNode.getNodeType() == Node.ELEMENT_NODE)
    {
        Element eElement = (Element) nNode;
        //vevő adatai
        System.out.println("vevo Id: " +
eElement.getAttribute("Vkod"));
        System.out.println("vevo neve: " +
        eElement.getElementsByTagName("nev").item(0).getTextContent());
        System.out.println("Vevo telefonszama: " +
        eElement.getElementsByTagName("tszam").item(0).getTextContent());
    }
}

```

```

        System.out.println("Vevo cime: " + "\n"+
            eElement.getElementsByTagName("varos").item(0).getTextContent()+"
"+eElement.getElementsByTagName("irszam").item(0).getTextContent()
            +",
"+eElement.getElementsByTagName("utca").item(0).getTextContent()+" "+
            eElement.getElementsByTagName("hszam").item(0).getTextContent()+". " );
    }
}

}
catch(Exception e){}
}
}

```

<terminated> DOMReadFHI2ET [Java Application]

Gyökér elem: boraszatok

boraszat Id: 001

Boraszat neve: Laboda

Boraszat alapitasa: 1997

Boraszat cime:

Miskolc 3515, Salétrom 14.

boraszat Id: 002

Boraszat neve: Uri

Boraszat alapitasa: 1996

Boraszat cime:

Csemő 3745, Vass 1.

tulajdonos Id: 100

Tulajdonos neve: Laboda Dániel

Tulajdonos születési ideje: 1970

Tulajdonos születési helye: Miskolc

tulajdonos Id: 200

Tulajdonos neve: Uri-Kovács Bence

Tulajdonos születési ideje: 1964

Tulajdonos születési helye: Csemő

bor Id: 010

Bor neve: Kóbor

Bor ara: 1000

Bor fajta: Tablettás

bor Id: 020

Bor neve: Dorombor

Bor ara: 2300

Bor fajta: Irsai

vevo Id: 111

vevo neve: Rőthi János

Vevo telefonszama: 06209365850

Vevo cime:

Kurityán 3650, Szabad 11.

vevo Id: 222

vevo neve: sirnakó Boldizsár

Vevo telefonszama: 06202345850

Vevo cime:

Szabadka 4250, Foglaltka 41.

2 b) Adat módosítás

```
package DOMreadFHI2ET;

import java.io.File;
import java.io.IOException;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.transform.OutputKeys;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerConfigurationException;
import javax.xml.transform.TransformerException;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.TransformerFactoryConfigurationError;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;
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 DOMModifyFHI2ET {
    public static void main(String[] args) {
        String filePath = "src/DOMreadFHI2ET/FHI2ET.xml";
        File xmlFile = new File(filePath);
        DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder;
        try {
            dBuilder = dbFactory.newDocumentBuilder();
            Document doc = dBuilder.parse(xmlFile);
            doc.getDocumentElement().normalize();

            // Módosítja a boraszat székhelyének címét
```

```

        updateElementValue(doc);

        // Törli a tulajdonos címét
        deleteElement(doc);

        // Hozzáad egy "tipus" elemet a borhoz
        addElement(doc);

        // Kiírja egy fileba a módosításokat
        writeXMLFile(doc);

    } catch (SAXException | ParserConfigurationException | IOException | TransformerException e1) {
        e1.printStackTrace();
    }
}

private static void writeXMLFile(Document doc)
throws TransformerFactoryConfigurationError, TransformerConfigurationException, TransformerException {
    doc.getDocumentElement().normalize();

    TransformerFactory transformerFactory = TransformerFactory.newInstance();
    Transformer transformer = transformerFactory.newTransformer();

    DOMSource source = new DOMSource(doc);
    StreamResult result = new StreamResult(new File("D:\\uj.xml"));
    transformer.setOutputProperty(OutputKeys.INDENT, "yes");
    transformer.transform(source, result);

    System.out.println("XML módosítása sikeres volt");
}

//Elem hozzáadása a bor-hoz.
private static void addElement(Document doc) {
    NodeList boraszatok = doc.getElementsByTagName("bor");
    Element bor = null;

    // loop for each user

```

```

for (int i = 0; i < boraszatok.getLength(); i++) {
    bor = (Element) boraszatok.item(i);

    Element id = doc.createElement("szarmazasi-hely");
    id.appendChild(doc.createTextNode("pokol-bugyra"));
    bor.appendChild(id);
}
}

```

//Egy elem törlése

```

private static void deleteElement(Document doc) {
    NodeList boraszatok = doc.getElementsByTagName("tulajdonos");
    Element tulajdonos = null;

    // loop for each user
    for (int i = 0; i < boraszatok.getLength(); i++) {
        tulajdonos = (Element) boraszatok.item(i);
        Node Szid = tulajdonos.getElementsByTagName("Szid").item(0);
        tulajdonos.removeChild(Szid);
    }
}

```

//elem frissítése az XML doksiban.

```

private static void updateElementValue(Document doc) {
    NodeList boraszatok = doc.getElementsByTagName("boraszat");
    Element boraszat = null;

    // loop for each user
    for (int i = 0; i < boraszatok.getLength(); i++) {
        boraszat = (Element) boraszatok.item(i);
        Node alap = boraszat.getElementsByTagName("alap").item(0).getFirstChild();
        alap.setNodeValue("9999");
    }
}
}

```



```

<bor Bkod="010">
    <nev>Kobor</nev>
    <ar>1000</ar>
    <ev>2001</ev>
    <fajta>Tablettas</fajta>
    <szarmazasi-hely>pokol-bugyra</szarmazasi-hely>
</bor>

<bor Bkod="020">
    <nev>Dorombor</nev>
    <ar>2300</ar>
    <ev>2021</ev>
    <fajta>Irsai</fajta>
    <szarmazasi-hely>pokol-bugyra</szarmazasi-hely>
</bor>

```

2 c) Adat lekérdezés:

```

package DOMreadFHI2ET;
import java.io.File;
import java.io.IOException;

import javax.sound.sampled.SourceDataLine;
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 DOMQueryFHI2ET {

    public static void main(String[] args) throws ParserConfigurationException,
SAXException, IOException {
        File file = new File("FHI2ET.xml");
        // Parse-olás
        DocumentBuilderFactory dbFactory =
DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();

```

```

Document doc = dBuilder.parse(file);
doc.getDocumentElement().normalize();
// Root element kiirasa
System.out.print("Gyöker elem: ");
System.out.println(doc.getDocumentElement().getNodeName());
NodeList nList = doc.getElementsByTagName("boraszatok");

// Minden pizzeria attribútum kiiratasa

for (int i = 0; i < nList.getLength(); i++) {
    Node node = nList.item(i);
    System.out.println("\nElem nev : " + node.getNodeName());
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element elem = (Element) node;
        NodeList nList2 = elem.getChildNodes();
        for (int j = 0; j < nList2.getLength(); j++) {
            Node node2 = nList2.item(j);
            if (node2.getNodeType() == Node.ELEMENT_NODE) {
                Element elem2 = (Element) node2;
                if (!node2.getNodeName().equals("Cim")) {

System.out.println(node2.getNodeName() + " : " + node2.getTextContent());
                } else {
                    System.out.println("Cim:");

                    NodeList nList3 =

elem2.getChildNodes();

nList3.getLength(); k++) {

                        Node node3 = nList3.item(k);

                        if (node3.getNodeType() ==

Node.ELEMENT_NODE) {

                                System.out.println("

" + node3.getNodeName() + " : " + node3.getTextContent());
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
//Kiirja annak a pizzerianak a nevet, ami Pesten van
System.out.println("\n Laboda nevu boraszat: \n");
for (int i = 0; i < nList.getLength(); i++) {
    Node node = nList.item(i);
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element elem = (Element) node;
        NodeList nList2 = elem.getChildNodes();
        for (int j = 0; j < nList2.getLength(); j++) {
            Node node2 = nList2.item(j);
            if (node2.getNodeType() == Node.ELEMENT_NODE) {
                Element elem2 = (Element) node2;
                NodeList nList3 = elem2.getChildNodes();
                for (int k = 0; k < nList3.getLength();

k++) {

                        Node node3 = nList3.item(k);

```

```
Node.ELEMENT_NODE) {
    if (node3.getNodeType() == Node.TEXT_NODE) {
        if (node3.getNodeName().equals("nev")) {
            if (node3.getTextContent().equals("Laboda")) {
                node2 = nList2.item(1);
                System.out.println(node2.getNodeName() + ": " + node2.getTextContent());
            }
        }
    }
}
}
```

kilistázza a dokumentumban lévő adatokat:

<terminated> DOMQueryFHI2E1 [Java Appli

Gyökér elem: boraszatok

Elem nev : boraszatok

boraszat :

Laboda

9999

Miskolc

3515

14

Saletrom

boraszat :

Uri

9999

Csemo

3745

1

Vass

bor :

Kobor

1000

2001

Tablettas

pokol-bugyra

Kilistázza a „Laboda” nevű borászatok adatait:

```
Problems @ Javadoc Declaration Cor
<terminated> DOMQueryFHI2ET [Java Applicatio

Laboda nevu boraszat:
boraszat:
    Laboda
    9999

    Miskolc
    3515
    14
    Saletrom
```