JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Készítette: **Sebák Petra**

Neptunkód: **FB8YPQ**

1. Feladat

A választott feladatom témája a tánc. Egy tánciskola gálájának előkészületeiben résztvevő személyekről, a kapcsolataikról és a gálához szükséges eszközökről szól.

Egy táncgála egy vagy több tánciskola által bemutatott koreográfiákról szól. Ehhez szükség van arra, hogy tudjuk pontosan hány és milyen tánciskolák lépnek fel. Miután ezt tudjuk, először is fontos, hogy kik fognak az adott táncgálán fellépni, tehát, hogy melyek azok a csoportok, akik részt fognak venni.

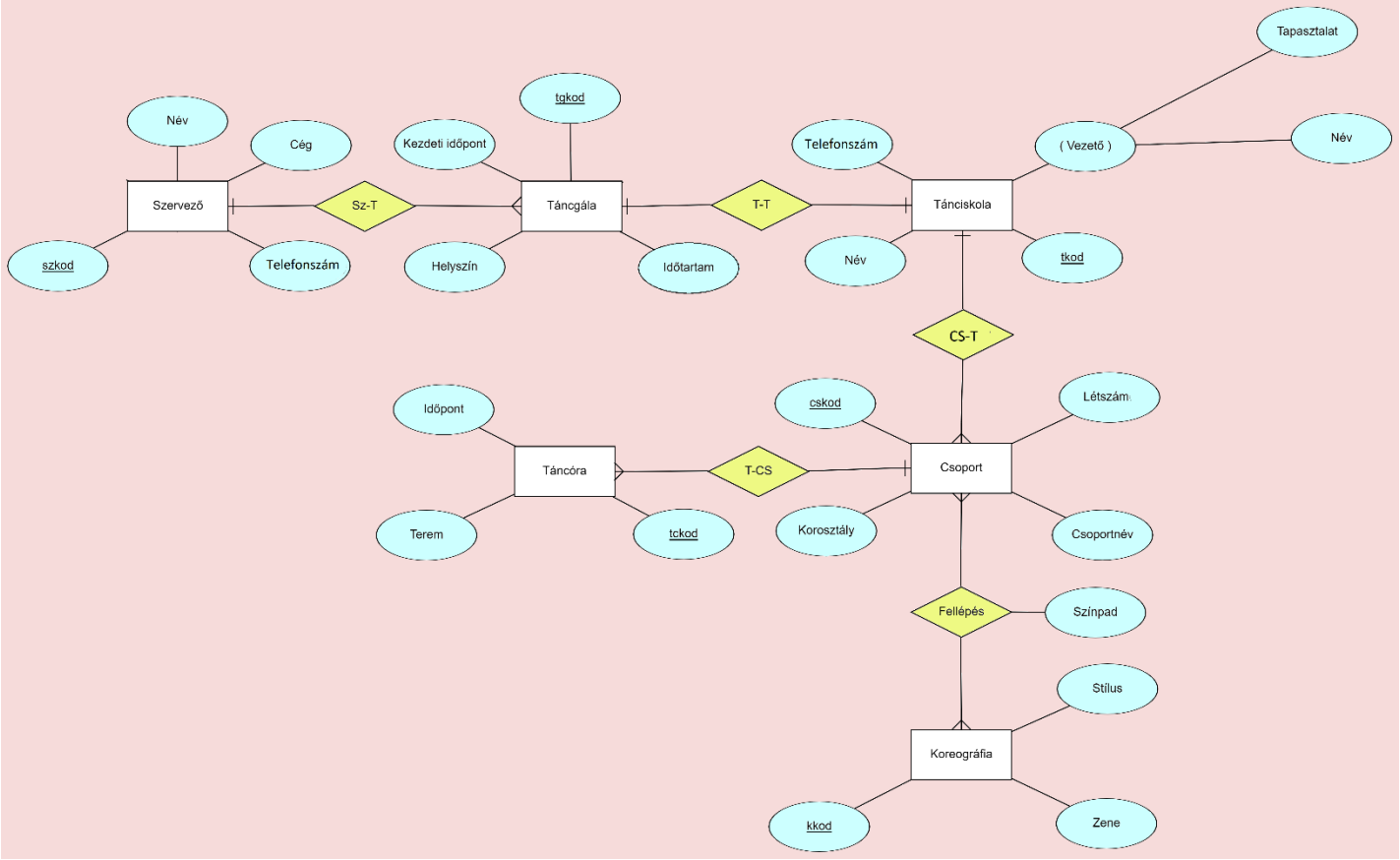
A fellépéshez szükséges minden csoportnak egy koreográfia, amelyet a gálán fog előadni. Ezeket a koreográfiákat a tánciskolával maga a fellépés köti össze és persze tegy csoport több koreográfiát is táncolhat és egy koreográfiát több csoport is megtanulhat.

A csoportok egy táncóra keretein belül tanulják meg a koreográfiákat, amelyeknek megvan a maga időpontja és helyszíne.
Fontos, hogy egy táncgála jól szervezett legyen, ezért szükség van egy szervezőre, aki koordinálja az embereket, megszervezi az eseményt.

Végül a legfontosabb maga a táncgála, aminek fontos tudnunk az időpontját és a helyszínét.

A kapcsolattartás és jól informáltság érdekében, mind a szervező, mind a tánciskola és az ott jelenlevő tanárok, diákok elérhetőségei fontosak, ezért ezeket az adatokat is beleraktam a feladatomba.

a) Az adatbázis ER modellje

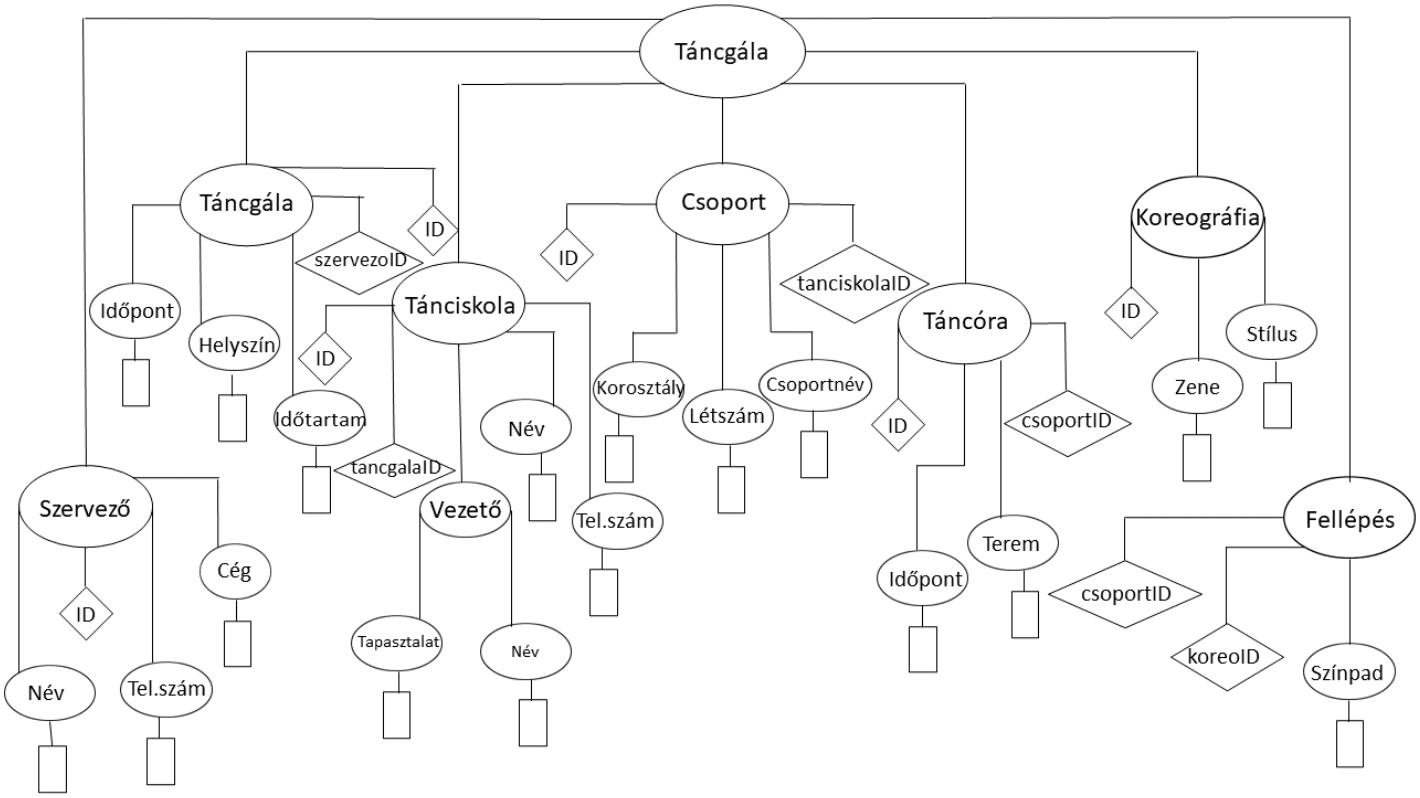


b) Az adatbázis konvertálása XDM modellre

A konvertálást követően idegenkulcsok jönnek létre. A Táncgála objektumban megjelenik az szervezoId, amely a szervezőre mutató idegenkulcs, illetve a Tánciskolánál megjelenik a tancgalaId, ami megmutatja, hogy az adott tánciskola mely táncgálán lép fel.

A csoport objektum tartalmazza, hogy mely az adott csoport melyik tánciskolához tartozik, ebben segít a tanciskolaId. A Táncóra objektumnál pedig feltűnik a csoportId, amely alapján megnézhetjük melyik csoport melyik táncórára jár.

A több-több kapcsolatból létrejön a Fellépés objektum, amely tartalmazza a kapcsolat tulajdonságát, illetve további két idegenkulcsot, amely a csoport és a koreográfia azonosítóját jelzi.



c) XML dokumentum

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><galaSzervezes
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="galaXMLSchemaFB8YPQ.xsd">
    <Fellepes id="1" koreoId="1" csoportId="1" >
        <szinpad>Kicsi</szinpad>
    </Fellepes>
    <Fellepes id="2" koreoId="2" csoportId="4" >
        <szinpad>Nagy</szinpad>
    </Fellepes>
    <Fellepes id="3" koreoId="2" csoportId="3" >
        <szinpad>Nagy</szinpad>
    </Fellepes>
    <Fellepes id="4" koreoId="3" csoportId="3" >
        <szinpad>Kicsi</szinpad>
    </Fellepes>
    <Fellepes id="5" koreoId="3" csoportId="2" >
        <szinpad>Kicsi</szinpad>
    </Fellepes>
    <Szervezo id="1">
        <Ceg>DanceHall Kft.</Ceg>
        <Nev>Nagy Fanni</Nev>
        <Telefonszam>22-111-45-21</Telefonszam>
    </Szervezo>
    <Tancgala id="1" szervezoId="1">
        <Helyszin>Miskolc</Helyszin>
        <Idopont>2021.01.30. 15:00</Idopont>
        <Idotartam>2 óra</Idotartam>
    </Tancgala>
```

```xml
<Tancgala id="2" szervezoId="1">
    <Helyszin>Debrecen</Helyszin>
    <Idopont>2022.04.12. 19:00</Idopont>
    <Idotartam>1 óra</Idotartam>
</Tancgala>
<Tanciskola id="1" tancgalaId="1">
    <TanciskolaNev>Elte</TanciskolaNev>
    <Telefonszam>21-377-20-69</Telefonszam>
    <Vezeto>
        <Nev>Tóth Evelin</Nev>
        <Tapasztalat>3 év</Tapasztalat>
    </Vezeto>
</Tanciskola>
<Tanciskola id="2" tancgalaId="2">
    <TanciskolaNev>Dance Elit</TanciskolaNev>
    <Telefonszam>21-343-22-88</Telefonszam>
    <Vezeto>
        <Nev>Hajdú Sándor</Nev>
        <Tapasztalat>1 év</Tapasztalat>
    </Vezeto>
</Tanciskola>
<Csoport id="1" tanciskolaId="1">
    <Korosztaly>9-11</Korosztaly>
    <Letszam>10</Letszam>
    <Csoportnev>CoolKids</Csoportnev>
</Csoport>
<Csoport id="2" tanciskolaId="1">
    <Korosztaly>12-14</Korosztaly>
    <Letszam>13</Letszam>
    <Csoportnev>Girls</Csoportnev>
</Csoport>
<Csoport id="3" tanciskolaId="1">
    <Korosztaly>15-17</Korosztaly>
    <Letszam>9</Letszam>
    <Csoportnev>Contemp</Csoportnev>
</Csoport>
<Csoport id="4" tanciskolaId="2">
    <Korosztaly>8-11</Korosztaly>
    <Letszam>19</Letszam>
    <Csoportnev>Unicorns</Csoportnev>
</Csoport>
<Csoport id="5" tanciskolaId="2">
    <Korosztaly>12-16</Korosztaly>
    <Letszam>5</Letszam>
    <Csoportnev>TroubleMakers</Csoportnev>
</Csoport>
<Tancora csoportId="1" id="1">
    <Idopont>Hétfő 14</Idopont>
    <Terem>3</Terem>
</Tancora>
<Tancora csoportId="2" id="2">
    <Idopont>Hétfő 16</Idopont>
    <Terem>3</Terem>
</Tancora>
```

```
    <Tancora csoportId="3" id="3">
        <Idopont>Kedd 14</Idopont>
        <Terem>3</Terem>
    </Tancora>
    <Tancora csoportId="4" id="4">
        <Idopont>Szerda 14</Idopont>
        <Terem>3</Terem>
    </Tancora>
    <Tancora csoportId="4" id="5">
        <Idopont>Péntek 18</Idopont>
        <Terem>3</Terem>
    </Tancora>
    <Koreografia id="1">
        <Zene>Adele-Rolling in the deep</Zene>
        <Stilus>HipHop</Stilus>
    </Koreografia>
    <Koreografia id="2">
        <Zene>Betti-Cry for help</Zene>
        <Stilus>Modern</Stilus>
    </Koreografia>
    <Koreografia id="3">
        <Zene>Jatleg-Égen át</Zene>
        <Stilus>HipHop</Stilus>
    </Koreografia>
</galaSzervezes>
```

d) XMLSchema készítése

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="galaSzervezes">
        <xs:complexType>
            <xs:sequence>
                <xs:choice maxOccurs="unbounded">
                    <xs:element name="Fellepes" type="fellepesType"/>
                    <xs:element name="Szervezo" type="szervezoType"/>
                    <xs:element name="Tancgala" type="tancgalaType"/>
                    <xs:element name="Tanciskola" type="tanciskolaType"/>
                    <xs:element name="Csoport" type="csoportType"/>
                    <xs:element name="Tancora" type="tancoraType"/>
                    <xs:element name="Koreografia" type="koreografiaType"/>
                </xs:choice>
            </xs:sequence>
        </xs:complexType>
        <xs:key name="koreoId_PK">
            <xs:selector xpath="Koreografia"/>
            <xs:field xpath="@id"/>
        </xs:key>
        <xs:key name="csoportId_PK">
            <xs:selector xpath="Csoport"/>
            <xs:field xpath="@id"/>
        </xs:key>
        <xs:key name="szervezoId_PK">
```

```
            <xs:selector xpath="Szervezo"/>
            <xs:field xpath="@id"/>
        </xs:key>
        <xs:key name="tanciskolaId_PK">
            <xs:selector xpath="Tanciskola"/>
            <xs:field xpath="@id"/>
        </xs:key>
        <xs:key name="tancgalaId_PK">
            <xs:selector xpath="Tancgala"/>
            <xs:field xpath="@id"/>
        </xs:key>
        <xs:keyref name="koreoId_FK" refer="koreoId_PK">
            <xs:selector xpath="Fellepes"/>
            <xs:field xpath="@koreoId"/>
        </xs:keyref>
        <xs:keyref name="csoportId_FK" refer="csoportId_PK">
            <xs:selector xpath="Fellepes"/>
            <xs:field xpath="@csoportId"/>
        </xs:keyref>
        <xs:keyref name="tanciskolaId_FK" refer="tanciskolaId_PK">
            <xs:selector xpath="Csoport"/>
            <xs:field xpath="@tId"/>
        </xs:keyref>
        <xs:keyref name="tancgalaId_FK" refer="tancgalaId_PK">
            <xs:selector xpath="Tanciskola"/>
            <xs:field xpath="@tancgalaId"/>
        </xs:keyref>
        <xs:keyref name="szervezoId_FK" refer="szervezoId_PK">
            <xs:selector xpath="Tancgala"/>
            <xs:field xpath="@szId"/>
        </xs:keyref>
    </xs:element>

    <xs:complexType name="fellepesType">
        <xs:sequence>
            <xs:element name="szinpad"/>
        </xs:sequence>
        <xs:attribute name="id" type="xs:unsignedByte" use="required"/>
        <xs:attribute name="koreoId" type="xs:unsignedByte" use="required"/>
        <xs:attribute name="csoportId" type="xs:unsignedByte" use="required"/>
    </xs:complexType>
    <xs:complexType name="szervezoType">
        <xs:sequence>
            <xs:element name="Ceg" type="xs:string"/>
            <xs:element name="Nev" type="xs:string"/>
            <xs:element name="Telefonszam" type="telType"/>
        </xs:sequence>
        <xs:attribute name="id" type="xs:unsignedByte" use="required"/>
    </xs:complexType>
    <xs:complexType name="tancgalaType">
        <xs:sequence>
            <xs:element name="Helyszin" type="xs:string"/>
            <xs:element name="Idopont" type="tancgalaIdopontType"/>
            <xs:element name="Idotartam" type="xs:string"/>
```

```
        </xs:sequence>
        <xs:attribute name="id" type="xs:unsignedByte" use="required"/>
        <xs:attribute name="szervezoId" type="xs:unsignedByte" use="required"/>
    </xs:complexType>
    <xs:complexType name="tanciskolaType">
        <xs:sequence>
            <xs:element name="TanciskolaNev" type="xs:string"/>
            <xs:element name="Telefonszam" type="telType"/>
            <xs:element name="Vezeto" type="vezetoType"/>
        </xs:sequence>
        <xs:attribute name="id" type="xs:unsignedByte" use="required"/>
        <xs:attribute name="tancgalaId" type="xs:unsignedByte" use="required"/>
    </xs:complexType>
    <xs:complexType name="csoportType">
        <xs:sequence>
            <xs:element name="Korosztaly" type="xs:string"/>
            <xs:element name="Letszam" type="xs:integer"/>
            <xs:element name="Csoportnev" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="id" type="xs:unsignedByte" use="required"/>
        <xs:attribute name="tanciskolaId" type="xs:unsignedByte" use="required"/>
    </xs:complexType>
    <xs:complexType name="tancoraType">
        <xs:sequence>
            <xs:element name="Idopont" type="tancoraIdopontType"/>
            <xs:element name="Terem" type="xs:integer"/>
        </xs:sequence>
        <xs:attribute name="id" type="xs:unsignedByte" use="required"/>
        <xs:attribute name="csoportId" type="xs:unsignedByte" use="required"/>
    </xs:complexType>
    <xs:complexType name="koreografiaType">
        <xs:sequence>
            <xs:element name="Zene" type="xs:string"/>
            <xs:element name="Stilus" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="id" type="xs:unsignedByte" use="required"/>
    </xs:complexType>
    <xs:complexType name="vezetoType">
        <xs:sequence>
            <xs:element name="Nev" type="xs:string"/>
            <xs:element name="Tapasztalat" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>

    <xs:simpleType name="tancoraIdopontType">
        <xs:restriction base="xs:string">
          <xs:pattern value="Hétfő [0-9][0-9]*|Kedd [0-9][0-9]*|Szerda [0-9][0-9]*|Csütörtök [0-
9][0-9]*|Péntek [0-9][0-9]*"/>
        </xs:restriction>
      </xs:simpleType>

      <xs:simpleType name="tancgalaIdopontType">
        <xs:restriction base="xs:string">
          <xs:pattern value="[0-9][0-9][0-9][0-9].[0-9][0-9].[0-9][0-9]. [0-9][0-9]*:[0-9][0-9]"/>
```

```
        </xs:restriction>
      </xs:simpleType>

    <xs:simpleType name="telType">
        <xs:restriction base="xs:string">
            <xs:pattern value="\d{2}-\d{3}-\d{2}-\d{2}"/>
            <xs:pattern value="\d{2}-\d{3}-\d{3}"/>
        </xs:restriction>
    </xs:simpleType>
</xs:schema>
```

2. Feladat

a) DOM JAVA olvasás kommentekkel

```
package hu.domparse.fb8ypq;

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;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import java.io.File;
import java.io.IOException;


public class DomReadFB8YPQ {

    public static void main(String[] args) {
        try {
            File xmlFile = new File("XMLFB8YPQ.xml"); // fájl, amiből olvasunk
            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance(); // XML
dokumentumból DOM objektum -
                                                                                    // lehetővé
tétele
            DocumentBuilder dBuilder = factory.newDocumentBuilder(); // XML fájl, Document
lekéréséhez
            Document doc = dBuilder.parse(xmlFile); // dokument lekérése
            doc.getDocumentElement().normalize();
            System.out.println("Táncgála adatok lekérése");
            Read(doc); // fő metódus, meghívódik a Read
        } catch (ParserConfigurationException pce) {
            pce.printStackTrace();
        } catch (IOException ioe) {
            ioe.printStackTrace();
```

```
        } catch (SAXException sae) {
            sae.printStackTrace();
        }
    }

    public static void Read(Document doc) {
        NodeList nList = doc.getElementsByTagName("Fellepes"); // Fellepes taggal rendelkező
elemek lekérése
                                                               // listába
        for (int i = 0; i < nList.getLength(); i++) { // listán végigmegyünk
            Node nNode = nList.item(i); // lekérjük a lista aktuális elemét, Elementé konvertáljuk
            Element element = (Element) nNode;
            // Lekérjük az attribútumokat, majd azok segítségével meghívjuk a definiált
            // metódusokat
            if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                String szinpad = element.getElementsByTagName("szinpad").item(0).getTextContent();
// darabszám lekérdezése
                String koreografiaId = element.getAttribute("koreoId");
                String csoportId = element.getAttribute("csoportId");
                System.out.println("\n-----------------------------------" + (i + 1)
                        + ". fellépés-----------------------------------");
                System.out.println("\n\tSzínpad:\t" + szinpad);
                ReadKoreografiaById(doc, koreografiaId);
                ReadCsoportById(doc, csoportId);
            }
        }
    }

    // fa struktúra miatt az attribútumban megadott id alapján kérdezzük le az egyes
    // rendeléshez tartozó elemeket
    // A legtöbb objektum rendelkezik leszármazottal, amelyet egy újabb metódus
    // kérdez le, az attrubútumban megadott ID alapján
    public static void ReadCsoportById(Document doc, String id) {
        NodeList nList = doc.getElementsByTagName("Csoport");
        for (int i = 0; i < nList.getLength(); i++) {
            Node nNode = nList.item(i);
            Element element = (Element) nNode;
            if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                if (element.getAttribute("id").equals(id)) {
                    String korosztaly =
element.getElementsByTagName("Korosztaly").item(0).getTextContent();
                    String letszam =
element.getElementsByTagName("Letszam").item(0).getTextContent();
                    String csoportnev =
element.getElementsByTagName("Csoportnev").item(0).getTextContent();
                    System.out.println("Csoport adatok: \n\tKorosztály:\t" + korosztaly +
"\n\tLétszám:\t" + letszam + "\n\tCsoportnév:\t"
                            + csoportnev);
                    String tanciskolaId = element.getAttribute("tanciskolaId");
                    ReadTanciskolaById(doc,tanciskolaId);
                }
            }
        }
    }
```

```
    public static void ReadTancoraById(Document doc, String id) {
        NodeList nList = doc.getElementsByTagName("Tancora");
        for (int i = 0; i < nList.getLength(); i++) {
            Node nNode = nList.item(i);
            Element element = (Element) nNode;
            if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                if (element.getAttribute("id").equals(id)) {
                    String idopont =
element.getElementsByTagName("Idopont").item(0).getTextContent();
                    String terem = element.getElementsByTagName("Terem").item(0).getTextContent();
                    String csoportId = element.getAttribute("csId");
                    System.out.println("Táncóra adatok: \n\tIdőpont:\t" + idopont + "\n\tTerem:\t"
+ terem);
                    ReadCsoportById(doc, csoportId);
                }
            }
        }
    }

    public static void ReadKoreografiaById(Document doc, String id) {
        NodeList nList = doc.getElementsByTagName("Koreografia");
        for (int i = 0; i < nList.getLength(); i++) {
            Node nNode = nList.item(i);
            Element element = (Element) nNode;
            if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                if (element.getAttribute("id").equals(id)) {
                    String zene = element.getElementsByTagName("Zene").item(0).getTextContent();
                    String stilus =
element.getElementsByTagName("Stilus").item(0).getTextContent();
                    System.out.println("Koreográfia adatok: \n\tZene:\t" + zene + "\n\tStílus:\t"
+ stilus);
                }
            }
        }
    }

    public static void ReadTancgalaById(Document doc, String id) {
        NodeList nList = doc.getElementsByTagName("Tancgala");
        for (int i = 0; i < nList.getLength(); i++) {
            Node nNode = nList.item(i);
            Element element = (Element) nNode;
            if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                if (element.getAttribute("id").equals(id)) {
                    String helyszin =
element.getElementsByTagName("Helyszin").item(0).getTextContent();
                    String idopont =
element.getElementsByTagName("Idopont").item(0).getTextContent();
                    String idotartam =
element.getElementsByTagName("Idotartam").item(0).getTextContent();
                    System.out.println("Tancgála adatok: \n\tHelyszín:\t" + helyszin +
"\n\tIdőpont:\t" + idopont + "\n\tIdotartam:\t"
                            + idotartam);
                    String szervezoId = element.getAttribute("szervezoId");
```

```
                    ReadSzervezoById(doc,szervezoId);
                }
            }
        }
    }

    public static void ReadSzervezoById(Document doc, String id) {
        NodeList nList = doc.getElementsByTagName("Szervezo");
        for (int i = 0; i < nList.getLength(); i++) {
            Node nNode = nList.item(i);
            Element element = (Element) nNode;
            if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                if (element.getAttribute("id").equals(id)) {
                    String nev = element.getElementsByTagName("Nev").item(0).getTextContent();
                    String ceg = element.getElementsByTagName("Ceg").item(0).getTextContent();
                    String telszam =
element.getElementsByTagName("Telefonszam").item(0).getTextContent();
                    System.out.println("Szervező adatok: \n\tNév:\t" + nev + "\n\tCég:\t" + ceg +
"\n\tTelefonszám:\t"
                            + telszam);
                }
            }
        }
    }

    public static void ReadTanciskolaById(Document doc, String id) {
        NodeList nList = doc.getElementsByTagName("Tanciskola");
        for (int i = 0; i < nList.getLength(); i++) {
            Node nNode = nList.item(i);
            Element element = (Element) nNode;
            if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                if (element.getAttribute("id").equals(id)) {
                    String tname =
element.getElementsByTagName("TanciskolaNev").item(0).getTextContent();
                    String telszam =
element.getElementsByTagName("Telefonszam").item(0).getTextContent();
                    String nev = element.getElementsByTagName("Nev").item(0).getTextContent();
                    String tapasztalat =
element.getElementsByTagName("Tapasztalat").item(0).getTextContent();
                    System.out.println("Tanciskola adatok: \n\tTánciskola név:\t" + tname +
"\n\tTelefonszám:\t" + telszam + "\n\tVezető adatai:\t \n\tNév:\t"
                            + nev + "\n\tTapasztalat:\t" + tapasztalat);
                    String tancgalaId = element.getAttribute("tancgalaId");
                    ReadTancgalaById(doc,tancgalaId);
                }
            }
        }
    }

}
```

b) DOM JAVA lekérdezés kommentekkel

```java
package hu.domparse.fb8ypq;

import java.io.File;
import java.io.IOException;
import java.util.Scanner;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerException;
import javax.xml.transform.TransformerFactory;
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 DomQueryFB8YPQ {

    public static void main(String[] args)
            throws ParserConfigurationException, IOException, SAXException, TransformerException {
        // TODO Auto-generated method stub
        File xmlFile = new File("XMLFB8YPQ.xml"); // xml fájl bekérése
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance(); // olvasás lehetővé
tétele
        DocumentBuilder dBuilder = factory.newDocumentBuilder();
        Document doc = dBuilder.parse(xmlFile);
        doc.getDocumentElement().normalize();

        System.out.println("Root element: " + doc.getDocumentElement().getNodeName());
        System.out.println("------------------------------");
        System.out.println("Az Elte tánciskola adatai: ");
        LoadTanciskolaQuery(doc);

    }

    public static void LoadTanciskolaQuery(Document doc) throws TransformerException {
        NodeList nodeList = doc.getElementsByTagName("Tanciskola"); //Tanciskola elemek listázása
        String tanciskola;
        Element element = null;
        Node nNode = null;
        for (int i = 0; i < nodeList.getLength(); i++) {
            nNode = nodeList.item(i);
            element = (Element) nNode;
            String nev = element.getElementsByTagName("TanciskolaNev").item(0).getTextContent();
```

```
            System.out.println((i + 1) + ") " + nev);

        }
        //Tánciskola választása
        System.out.println("Írja be annak a tánciskolának a nevét, amelyikbe járó csoportok
adatait szeretné látni:");
        Scanner sc = new Scanner(System.in);
        tanciskola = sc.nextLine();
        for (int i = 0; i < nodeList.getLength(); i++) {
            nNode = nodeList.item(i);
            element = (Element) nNode;
            if (nNode.getNodeType() == Node.ELEMENT_NODE) {

                if (tanciskola.equals("Elte")) {
                    LoadCsoportQuery(doc, "1");
                    break;
                }

                if (tanciskola.equals("Dance Elit")) {
                    LoadCsoportQuery(doc, "2");
                    break;
                }
            }
        }

    }
    //Kiválasztott tánciskola csoportjai adatainak kiíratása
    public static void LoadCsoportQuery(Document doc, String id) throws TransformerException {
        NodeList nodeList = doc.getElementsByTagName("Csoport");
        int csoport = 0;

        for (int i = 0; i < nodeList.getLength(); i++) {
            Node nNode = nodeList.item(i);
            Element element = (Element) nNode;
            String tId = element.getAttribute("tanciskolaId");
            if (nNode.getNodeType() == Node.ELEMENT_NODE) {

                if (id.equals(tId)) {
                    csoport += 1;
                    System.out.println(csoport + ". csoport adatai:");
                    String csId = element.getAttribute("id");
                    DomReadFB8YPQ.ReadCsoportById(doc, csId);

                }
            }
        }

    }

}
```

c) DOM JAVA módosítás kód kommentekkel

```
package hu.domparse.fb8ypq;

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;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerException;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;
import java.io.File;
import java.io.IOException;
import java.util.Scanner;

public class DomModifyFB8YPQ {

    public static void main(String[] args) throws ParserConfigurationException, IOException,
SAXException, TransformerException {
         File xmlFile = new File("XMLFB8YPQ.xml"); //xml fájl bekérése
         DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance(); //olvasás lehetővé
tétele
         DocumentBuilder dBuilder = factory.newDocumentBuilder();
         Document doc = dBuilder.parse(xmlFile);
         doc.getDocumentElement().normalize();
         System.out.println("XML Módosítása");
         System.out.println("Adja meg mit szeretne módosítani: ");
         System.out.println("1 Táncgála módosítása\n2 Csoport módosítása\n3 Táncóra módosítása\n4
Szervező módosítása");
         Modify(doc);
         }

    public static void ModifyXML(Document doc) throws TransformerException {
        TransformerFactory transformerFactory = TransformerFactory.newInstance();
        Transformer transformer = transformerFactory.newTransformer();
        DOMSource source = new DOMSource(doc);
        StreamResult result = new StreamResult(new File("XMLFB8YPQ.xml"));
        transformer.transform(source, result);
    }

    // Modify-ban a felhasználót megkérdezzük, hogy a táncgála mely adatát kívánja
    // módosítani
    public static void Modify(Document doc) throws TransformerException {
        int tancgalakSzama = doc.getElementsByTagName("Tancgala").getLength(); // tancgalak
számának lekérdezése
```

```
        int csoportokSzama = doc.getElementsByTagName("Csoport").getLength(); // csoportok
számának lekérdezése
        int tancorakSzama = doc.getElementsByTagName("Tancora").getLength(); // tancorak számának
lekérdezése
        int szervezokSzama = doc.getElementsByTagName("Szervezo").getLength(); // szervezok
számának
                                                                                //
lekérdezése
        Scanner scan = new Scanner(System.in);
        System.out.println("Adja meg a sorszamot: ");
        int readCategory = scan.nextInt();
        switch (readCategory) {
        case 1:
            ModifyTancgala(doc, tancgalakSzama);
            break;
        case 2:
            ModifyCsoport(doc, csoportokSzama);
            break;
        case 3:
            ModifyTancora(doc, tancorakSzama);
            break;
        case 4:
            ModifySzervezo(doc, szervezokSzama);
            break;
        }
    }

    private static void ModifyTancgala(Document doc, int tancgalaszam) throws TransformerException
{
        // Kiiratjuk a jelenlegi táncgálákat, majd lekérdezzük melyiket kívánja
        // módosítani.
        System.out.println("Melyik táncgála adatait szeretné módosítani?");
        for (int i = 1; i < tancgalaszam + 1; i++) {
            System.out.println(i + ". táncgála");
            DomReadFB8YPQ.ReadTancgalaById(doc, String.valueOf(i));
            System.out.println("----------------------------------------");
        }
        String id = ReadId();
        // Bekérjük az új adatokat
        Scanner sc = new Scanner(System.in);
        System.out.print("Helyszín: ");
        String helyszin = sc.nextLine();
        System.out.print("Időpont: ");
        String idopont = sc.nextLine();
        System.out.print("Időtartam: ");
        String idotartam = sc.nextLine();
        // lekérdezzük az Elementeket, majd setTextContent-el módosítjuk.
        NodeList nodeList = doc.getElementsByTagName("Tancgala");
        for (int i = 0; i < nodeList.getLength(); i++) {
            Node nNode = nodeList.item(i);
            if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                Element element = (Element) nNode;
                String sid = element.getAttribute("id");
                if (sid.equals(id)) {
```

```
                Node node1 = element.getElementsByTagName("Helyszin").item(0);
                node1.setTextContent(helyszin);
                Node node2 = element.getElementsByTagName("Idopont").item(0);
                node2.setTextContent(idopont);
                Node node3 = element.getElementsByTagName("Idotartam").item(0);
                node3.setTextContent(idotartam);
                System.out.println("Sikeres módosítás");
            }
        }
    }
    ModifyXML(doc); // Létrehozzuk az XML-t
}

private static void ModifyCsoport(Document doc, int csoportszam) throws TransformerException {
    System.out.println("Melyik csoportot kívánja módosítani?");
    for (int i = 1; i < csoportszam + 1; i++) {
        System.out.println(i + ". csoport");
        DomReadFB8YPQ.ReadCsoportById(doc, String.valueOf(i));
        System.out.println("----------------------------------------");
    }
    String id = ReadId();
    // Bekérjük az új adatokat
    Scanner sc = new Scanner(System.in);
    System.out.print("Korosztály: ");
    String korosztaly = sc.nextLine();
    System.out.print("Létszám: ");
    String letszam = sc.nextLine();
    System.out.print("Csoportnév: ");
    String csoportnev = sc.nextLine();
    // lekérdezzük az Elementeket, majd setTextContent-el módosítjuk.
    NodeList nodeList = doc.getElementsByTagName("Csoport");
    for (int i = 0; i < nodeList.getLength(); i++) {
        Node nNode = nodeList.item(i);
        if (nNode.getNodeType() == Node.ELEMENT_NODE) {
            Element element = (Element) nNode;
            String sid = element.getAttribute("id");
            if (sid.equals(id)) {
                Node node1 = element.getElementsByTagName("Korosztaly").item(0);
                node1.setTextContent(korosztaly);
                Node node2 = element.getElementsByTagName("Letszam").item(0);
                node2.setTextContent(letszam);
                Node node3 = element.getElementsByTagName("Csoportnev").item(0);
                node3.setTextContent(csoportnev);
                System.out.println("Sikeres módosítás");
            }
        }
    }
    ModifyXML(doc); // Létrehozzuk az XML-t
}

private static void ModifyTancora(Document doc, int tancoraszam) throws TransformerException {
    System.out.println("Melyik táncórát szeretné módosítani?");
    for (int i = 1; i < tancoraszam + 1; i++) {
        System.out.println(i + ". táncóra");
```

```
            DomReadFB8YPQ.ReadTancoraById(doc, String.valueOf(i));
            System.out.println("----------------------------------------");
        }
        String id = ReadId();
        Scanner sc = new Scanner(System.in);
        System.out.print("Időpont");
        String idopont = sc.nextLine();
        System.out.print("Terem: ");
        String terem = sc.nextLine();
        NodeList nodeList = doc.getElementsByTagName("Tancora");
        for (int i = 0; i < nodeList.getLength(); i++) {
            Node nNode = nodeList.item(i);
            if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                Element element = (Element) nNode;
                String sid = element.getAttribute("id");
                if (sid.equals(id)) {
                    Node node1 = element.getElementsByTagName("Idopont").item(0);
                    node1.setTextContent(idopont);
                    Node node2 = element.getElementsByTagName("Terem").item(0);
                    node2.setTextContent(terem);
                    System.out.println("Sikeres módosítás");
                }
            }
        }
        ModifyXML(doc);
    }

    public static String ReadId() {
        Scanner sc = new Scanner(System.in);
        System.out.print("\nid:");
        String id = sc.nextLine();
        return id;
    }

    private static void ModifySzervezo(Document doc, int szervezoszam) throws TransformerException
{
        System.out.println("Melyik szervező adatait szeretné módosítani?");
        for (int i = 1; i < szervezoszam+1; i++) {
            System.out.println(i + ". specifikacio");
            DomReadFB8YPQ.ReadSzervezoById(doc, String.valueOf(i));
            System.out.println("----------------------------------------");
        }
        String id = ReadId();
        Scanner sc = new Scanner(System.in);
        System.out.print("Név: ");
        String nev = sc.nextLine();
        System.out.print("Cég: ");
        String ceg = sc.nextLine();
        System.out.print("Telefonszám: ");
        String telszam = sc.nextLine();
        NodeList nodeList = doc.getElementsByTagName("Szervezo");
        for (int i = 0; i < nodeList.getLength(); i++) {
            Node nNode = nodeList.item(i);
            if (nNode.getNodeType() == Node.ELEMENT_NODE) {
```

```
                Element element = (Element) nNode;
                String sid = element.getAttribute("id");
                if (sid.equals(id)) {
                    Node node1 = element.getElementsByTagName("Nev").item(0);
                    node1.setTextContent(nev);
                    Node node2 = element.getElementsByTagName("Ceg").item(0);
                    node2.setTextContent(ceg);
                    Node node3 = element.getElementsByTagName("Telefonszam").item(0);
                    node3.setTextContent(telszam);
                    System.out.println("Sikeres módosítás");
                }
            }
        }
        ModifyXML(doc);
    }
}
```