

JEGYZŐKÖNYV

Adatkezelés XML környezetben
Féléves feladat

Készítette: Kovács László

Neptunkód: M9QNOF

A feladat leírása:

Ez a jegyzőkönyv az Adatkezelés XML Környezetben tárgyhoz készült beadandóhoz készült, melynek témája egy versenyzőket nyilvántartó rendszer.

Ehhez kezdésnek el kellett készíteni egy ER modellt.(ER.png)

Az egyedek: Versenyző, Edző, Meccs illetve Szponzor.

A Versenyző – Szponzor, Versenyző – Meccs között N:M kapcsolat áll fent.

A Versenyző – Edző között N:1 kapcsolat van.

Ez az Egyed kapcsolat modell alapjául szolgált a következő modellnek, ami nem más mint az XDM modell.(XDM.png)

Amint ezzel kész voltam, ezek alapján elkészítettem az XML dokumentumot.(XMLM9QNOF.xml)

Majd amint az elvárásaimnak eleget tett az XML dokumentum, úgy egyből ugrottam is át az XMLSchema (XSD) készítésére, így amint azzal is kész voltam, az XML fájlom validálva lett. (XMLM9QNOF.xsd)

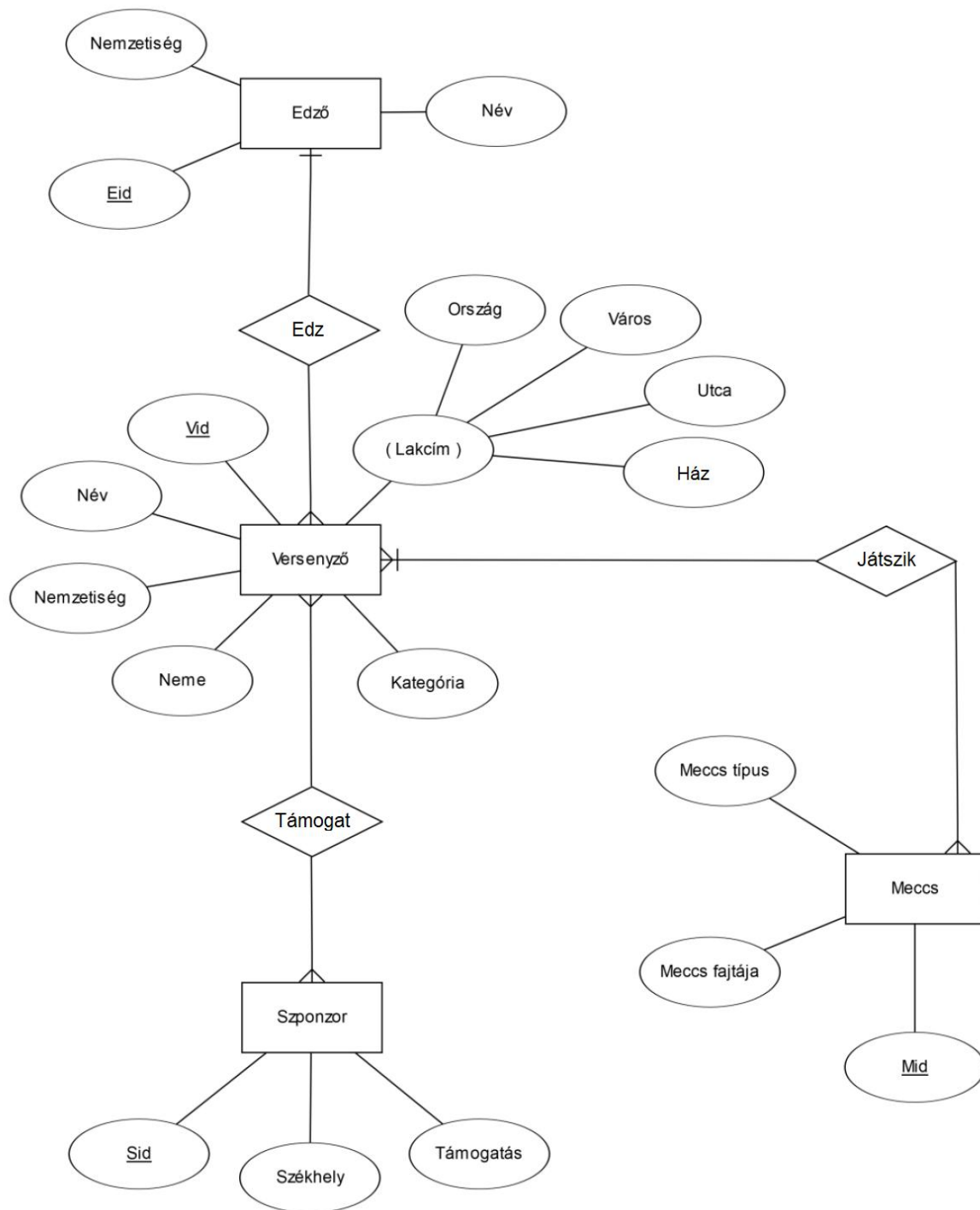
Ezek után kellett létrehoznom egy olvasó egy módosító DOM fájlt.

Ezeknél a fájloknál az órán tanultak alapján igyekeztem programozni.

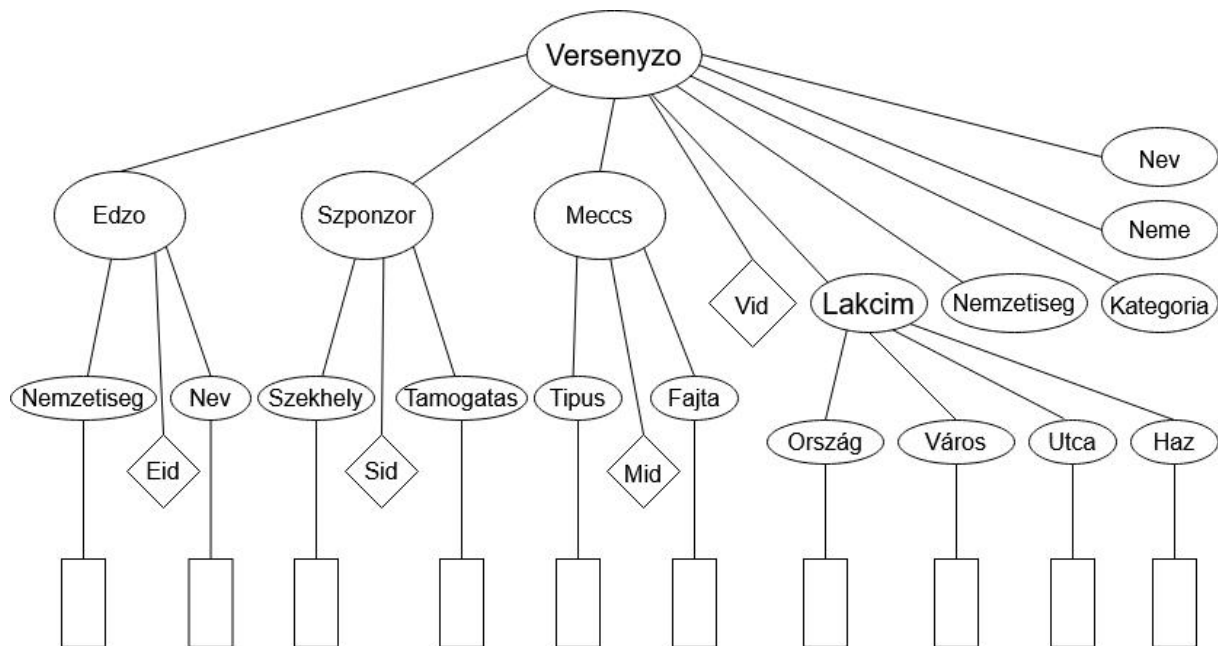
Az olvasó programban kiírtam a gyökérelemet, majd a versenyző(k) azonosítása után egy ciklusban node-ok segítségével kértem be, majd írtam ki az elemeket. Ugyanezen logika alkalmazásával írtam 3 metódust ami segítségével szebbé tehettem a kiírást. (DOMReadM9QNOF.java)

A módosító programban pedig hasonló elven haladtam, illetve a transfromerfactory segítségével sikerült az újonnan létrehozott (XMLM9QNOFMODIFIED) fájlban változtatást végezni. (DOMModifyM9QNOF.java)

1a)Az adatbázis ER modell:



1b)Az adatbázis konvertálása XDM modellre:



1c)Az XDM modell alapján XML dokumentum készítése:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-model href="XMLSchemaM9QNOF.xsd" type="application/xml"
schematypens="http://www.w3.org/2001/XMLSchema"?>

<versenyzok>
<versenyzo Vid="V_009">
  <nev>Bajnok Béla</nev>
  <nemzetiseg>Magyar</nemzetiseg>
  <nem>Férfi</nem>
  <kategoria>Pehelysúly</kategoria>
  <lakcim>
    <ország>Magyarország </ország><varos>Alsóbögyörősválagpuszta
  </varos><utca>Kő utca </utca><haz>14</haz>
  </lakcim>
  <edzo Eid="af_1117" Vid="V_009">
    <nemzetiseg>Szerb</nemzetiseg>
    <nev>Brc Zrc</nev>
  </edzo>
  <szponzorok>
    <szponzor Sid="Ferrari" Vid="V_009">
      <szekhely>Torino</szekhely>
      <tamogatas>1000000 EUR</tamogatas>
    </szponzor>
    <szponzor Sid="Pottyos" Vid="V_009">
      <szekhely>Budapest</szekhely>
      <tamogatas>11000000 HUF</tamogatas>
    </szponzor>
    <szponzor Sid="Milka" Vid="V_009">
      <szekhely>Bécs</szekhely>
    </szponzor>
  </szponzorok>
</versenyzo>
</versenyzok>
```

```

        <tamogatas>230000 EUR</tamogatas>
    </szponzor>
    <szponzor Sid="Nissin" Vid="V_009">
        <szekhely>Tokió</szekhely>
        <tamogatas>5000000000 JEN</tamogatas>
    </szponzor>
</szponzorok>

<meccsek>
    <meccs Mid="M55" Vid="V_009">
        <tipus>Box</tipus>
        <fajta>12 menet</fajta>
    </meccs>
    <meccs Mid="M56" Vid="V_009">
        <tipus>Box</tipus>
        <fajta>6 menet</fajta>
    </meccs>
    <meccs Mid="M57" Vid="V_009">
        <tipus>Box</tipus>
        <fajta>9 menet</fajta>
    </meccs>
    <meccs Mid="M58" Vid="V_009">
        <tipus>Box</tipus>
        <fajta>12 menet</fajta>
    </meccs>
</meccsek>
</versenyzo>

<versenyzo Vid="V_010" >
    <nev>Kevésbábajnok Kálmán</nev>
    <nemzetiseg>Magyar</nemzetiseg>
    <nem>Férfi</nem>
    <kategoria>Középsúly</kategoria>
    <lakcim>
        <ország>Magyarország </ország><varos>Tard </varos><utca>Pelyhes utca
    </utca><haz>59</haz>
    </lakcim>
    <edzo Eid="af_1117" Vid="V_010">
        <nemzetiseg>Szerb</nemzetiseg>
        <nev>Brc Zrc</nev>
    </edzo>
    <szponzorok>
        <szponzor Sid="Lamborghini" Vid="V_010">
            <szekhely>Velençe</szekhely>
            <tamogatas>30000 EUR</tamogatas>
        </szponzor>
        <szponzor Sid="JohnDeere" Vid="V_010">
            <szekhely>Wasghinton</szekhely>
            <tamogatas>19000 USD</tamogatas>
        </szponzor>
        <szponzor Sid="SaulTechnologies" Vid="V_010">
            <szekhely>Varsó</szekhely>
            <tamogatas>60000000 zlotyi</tamogatas>
        </szponzor>
        <szponzor Sid="Cunda" Vid="V_010">
            <szekhely>Bécs</szekhely>
            <tamogatas>50000 EUR</tamogatas>
        </szponzor>
    </szponzorok>

```

```

    <meccsek>
      <meccs Mid="M3044" Vid="V_010">
        <tipus>Box</tipus>
        <fajta>12 menet</fajta>
      </meccs>
      <meccs Mid="M3009" Vid="V_010">
        <tipus>Box</tipus>
        <fajta>8 menet</fajta>
      </meccs>
      <meccs Mid="M155" Vid="V_010">
        <tipus>Box</tipus>
        <fajta>12 menet</fajta>
      </meccs>
      <meccs Mid="M146" Vid="V_010">
        <tipus>Box</tipus>
        <fajta>10 menet</fajta>
      </meccs>
    </meccsek>
  </versenyzok>
</versenyzok>

```

1d) Az XML dokumentum alapján az XMLSchema készítése:

```

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

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="versenyzok">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="versenyzo" maxOccurs="unbounded"
type="vtype"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:complexType name="vtype">
    <xs:sequence>
      <xs:element name="nev"
type="xs:string" maxOccurs="1"/>
      <xs:element name="nemzetiseg"
type="xs:string" maxOccurs="1"/>
      <xs:element name="nem"
type="xs:string" maxOccurs="1"/>
      <xs:element name="kategoria"
type="xs:string" maxOccurs="1"/>
      <xs:element name="lakcim"
type="ltype" maxOccurs="1"/>
      <xs:element name="edzo" id="Eid"
maxOccurs="1"/>
      <xs:element name="szponzorok" maxOccurs="1">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="szponzor"
type="sztype" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

        <xs:element name="mecssek" maxOccurs="1">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="meccs" type="mtype"
maxOccurs="unbounded"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
    <xs:attribute name="Vid" type="xs:ID"/>
</xs:complexType>

    <xs:complexType name="sztype">
        <xs:sequence>
            <xs:element name="szekhely" type="xs:string"/>
            <xs:element name="tamogatas" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="Sid" type="xs:ID"/>
        <xs:attribute name="Vid" type="xs:IDREF"/>
    </xs:complexType>

<xs:complexType name="mtype">
    <xs:sequence>
        <xs:element name="tipus" type="xs:string"/>
        <xs:element name="fajta" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="Mid" type="xs:ID"/>
    <xs:attribute name="Vid" type="xs:IDREF"/>
</xs:complexType>

<xs:complexType name="etype">
    <xs:sequence>
        <xs:element name="nev" type="xs:string"/>
        <xs:element name="nemzetiseg" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="Eid" type="xs:ID" use="required"/>
    <xs:attribute name="Vid" type="xs:IDREF" use="optional"/>
</xs:complexType>

<xs:complexType name="ltype">
    <xs:sequence>
        <xs:element name="ország" type="xs:string" maxOccurs="1"/>
        <xs:element name="varos" type="xs:string" maxOccurs="1"/>
        <xs:element name="utca" type="xs:string" maxOccurs="1"/>
        <xs:element name="haz" type="xs:string" maxOccurs="1" />
    </xs:sequence>
</xs:complexType>

</xs:schema>

```

2a) Adatolvasás

```

package hu.domparse.mq9nof;
import java.io.*;
import javax.xml.parsers.*;

```

```

import org.w3c.dom.*;
import org.xml.sax.*;
public class DOMReadM9QNOF {
    public static void main(String[] args) throws SAXException, IOException,
ParserConfigurationException {
        //Létrehozás
        File xmlFile = new File("src/hu/domparse/mq9nof/XMLM9QNOF.xml");
        //Builder + konvertálás
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = factory.newDocumentBuilder();
        Document doc = dBuilder.parse(xmlFile);
        //Normalizálás
        doc.getDocumentElement().normalize();
        //Gyökérelem kiírása
        System.out.println("Gyökérelem: " +
doc.getDocumentElement().getNodeName());
        //versenyzo elemek nList-be illesztése
        NodeList nList = doc.getElementsByTagName("versenyzo");
        System.out.println("-----");
        --");
        //Kiírás ciklusa; nList/nNode-on végighaladva az elkészített
algoritmusokkal kiírja a kért adatokat
        for (int i = 0; i < nList.getLength(); i++) {
            Node nNode = nList.item(i);
            if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                Element elem = (Element) nNode;
                String Vid = elem.getAttribute("Vid");
                //Adatok átadása a node-okba
                Node node1 = elem.getElementsByTagName("nev").item(0);
                String nev = node1.getTextContent();
                Node node2 = elem.getElementsByTagName("nemzetiseg").item(0);
                String nemz = node2.getTextContent();
                Node node3 = elem.getElementsByTagName("nem").item(0);
                String nem = node3.getTextContent();
                Node node4 = elem.getElementsByTagName("kategoria").item(0);
                String kat = node4.getTextContent();
                Node node5 = elem.getElementsByTagName("lakcim").item(0);
                String lak = node5.getTextContent();
                //Kiírás
                System.out.println("Versenyző ID: " + Vid);
                System.out.println("Neve: " + nev);
                System.out.println("Nemzetiség: " + nemz);
                System.out.println("Neme: " + nem);
                System.out.println("Kategória: " + kat);
                System.out.println("Lakhelye " + lak);
                //Metódus hívások
                listEdzo(doc, Vid);
                System.out.println(nev + " szponzorai:\n" );
                listSzponzor(doc, Vid);
                System.out.println(nev + " Meccsei:\n" );
                listMeccsek(doc, Vid);
                System.out.println("\n");
            }
        }
        //Edzők adatainak kiírása az előző módszer alapján
        public static void listEdzo (Document doc, String Vid) {
            NodeList nList = doc.getElementsByTagName("edzo");
            for (int i = 0; i < nList.getLength(); i++) {

```



```

        Node nNode = nList.item(i);
        if (nNode.getNodeType() == Node.ELEMENT_NODE) {
            Element elem = (Element) nNode;
            if (elem.getAttribute("Vid").toString().equals(Vid)) {
                Node node1 =
elem.getElementsByTagName("nemzetiseg").item(0);
                String nemzetiseg = node1.getTextContent();
                Node node2 = elem.getElementsByTagName("nev").item(0);
                String nev = node2.getTextContent();
                System.out.println("Edző nemzetisége: " + nemzetiseg);
                System.out.println("Edző neve: " + nev + "\n");
            }
        }
    }
}

//Szponzorok adatainak kiírása az előző módszer alapján
public static void listSzponzor (Document doc, String Vid) {
    NodeList nList = doc.getElementsByTagName("szponzor");
    for (int i = 0; i < nList.getLength(); i++) {
        Node nNode = nList.item(i);
        if (nNode.getNodeType() == Node.ELEMENT_NODE) {
            Element elem = (Element) nNode;
            if (elem.getAttribute("Vid").toString().equals(Vid)) {
                String Sid = elem.getAttribute("Sid");
                Node node1 =
elem.getElementsByTagName("szekhely").item(0);
                String szekhely = node1.getTextContent();
                Node node2 =
elem.getElementsByTagName("tamogatas").item(0);
                String tamogatas = node2.getTextContent();
                System.out.println("Szponzor neve: " + Sid);
                System.out.println("Székhelye: " + szekhely);
                System.out.println("Támogatás összege: " + tamogatas +
"\n");
            }
        }
    }
}

//Meccsek adatainak kiírása az előző módszer alapján
public static void listMeccsek (Document doc, String Vid) {
    NodeList nList = doc.getElementsByTagName("meccs");
    for (int i = 0; i < nList.getLength(); i++) {
        Node nNode = nList.item(i);
        if (nNode.getNodeType() == Node.ELEMENT_NODE) {
            Element elem = (Element) nNode;
            if (elem.getAttribute("Vid").toString().equals(Vid)) {
                String Mid = elem.getAttribute("Mid");
                Node node1 = elem.getElementsByTagName("tipus").item(0);
                String tipus = node1.getTextContent();
                Node node2 = elem.getElementsByTagName("fajta").item(0);
                String fajta = node2.getTextContent();
                System.out.println("Meccs ID: " + Mid);
                System.out.println("Meccs típusa: " + tipus);
                System.out.println("Meccs fajtája: " + fajta + "\n");
            }
        }
    }
}
}

```

2b) Adatmódosítás

```
package hu.domparse.mq9nof;
import java.io.*;
import java.util.Scanner;
import javax.xml.parsers.*;
import javax.xml.transform.*;
import javax.xml.transform.dom.*;
import javax.xml.transform.stream.*;
import org.w3c.dom.*;
import org.xml.sax.*;
public class DOMModifyM9QNOF {
    public static void main(String[] args) throws SAXException, IOException,
    ParserConfigurationException, TransformerConfigurationException,
    TransformerException {
        //Az eredeti XML fájl és a megváltoztatott xml fájl megadása
        File xmlFile = new File("src/hu/domparse/mq9nof/XMLM9QNOF.xml");
        File xmlFileMODIFIED = new
    File("src/hu/domparse/mq9nof/XMLM9QNOFMODIFIED.xml");
        //Scanner nyitása a beolvasáshoz
        Scanner sc = new Scanner(System.in);
        //Builder + konvertálás
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = factory.newDocumentBuilder();
        Document doc = dBuilder.parse(xmlFile);
        //Normalizálás
        doc.getDocumentElement().normalize();
        //A versenyző elemek kiválasztása.
        NodeList nList = doc.getElementsByTagName("versenyzo");
        //ciklus a változtatásokhoz
        for (int i = 0; i < nList.getLength(); i++) {
            Node nNode = nList.item(i);
            Element elem = (Element) nNode;
            //Név bekérése
            Node node1 = elem.getElementsByTagName("nev").item(0);
            String nev = node1.getTextContent();
            //Átadás a node-nak
            System.out.println("A versenyző jelenlegi neve: " + nev + "\n");
            System.out.println("Add meg a versenyző új nevét: \n");
            //Bekérés billentyűzeten
            String modifiedname = sc.next();
            //Beállítás node-on keresztül
            node1.setTextContent(modifiedname);
        }
        //Scanner zárása
        sc.close();
        //transformer és domsource használatával változtatjuk a fájlt
        TransformerFactory transformerFactory = TransformerFactory.newInstance();
        Transformer transformer = transformerFactory.newTransformer();
        DOMSource source = new DOMSource(doc);
        //A változtatás a result-ba került
        StreamResult result = new StreamResult(xmlFileMODIFIED);
        //Beírásra kerül a módosított fájlba a módosítás
        transformer.transform(source, result);
    }
}
```