

JEGYZŐKÖNYV

ADATKEZELÉS XML KÖRNYEZETBEN
FÉLÉVES FELADAT

Készítette: Schneider Henrik
Neptun kód: NRTHZZ

A feladat leírása:

Az adatkezelés XML környezetben féléves feladat során egy kórház E-R adatbázist hoztam létre. A beadandó tematikája a pácienseket és a hozzájuk tartozó adatokat írja le. A létrehozandó XML dokumentumban a játékosok adatain kívül nyilván lesznek tartva a pácienseket műtő orvosok, a műtők, a gyógyszerek amiket szednek.

A modellben találhatóak összetett, többértékű és opcionális tulajdonságok, melyek az alábbi dokumentumokban is megfelelően modellezésre kerültek.

A modellben található vegyesen 1:1 1:N és N:M tulajdonságok is. Az N:M tulajdonság pedig rendelkezik saját attribútummal is.

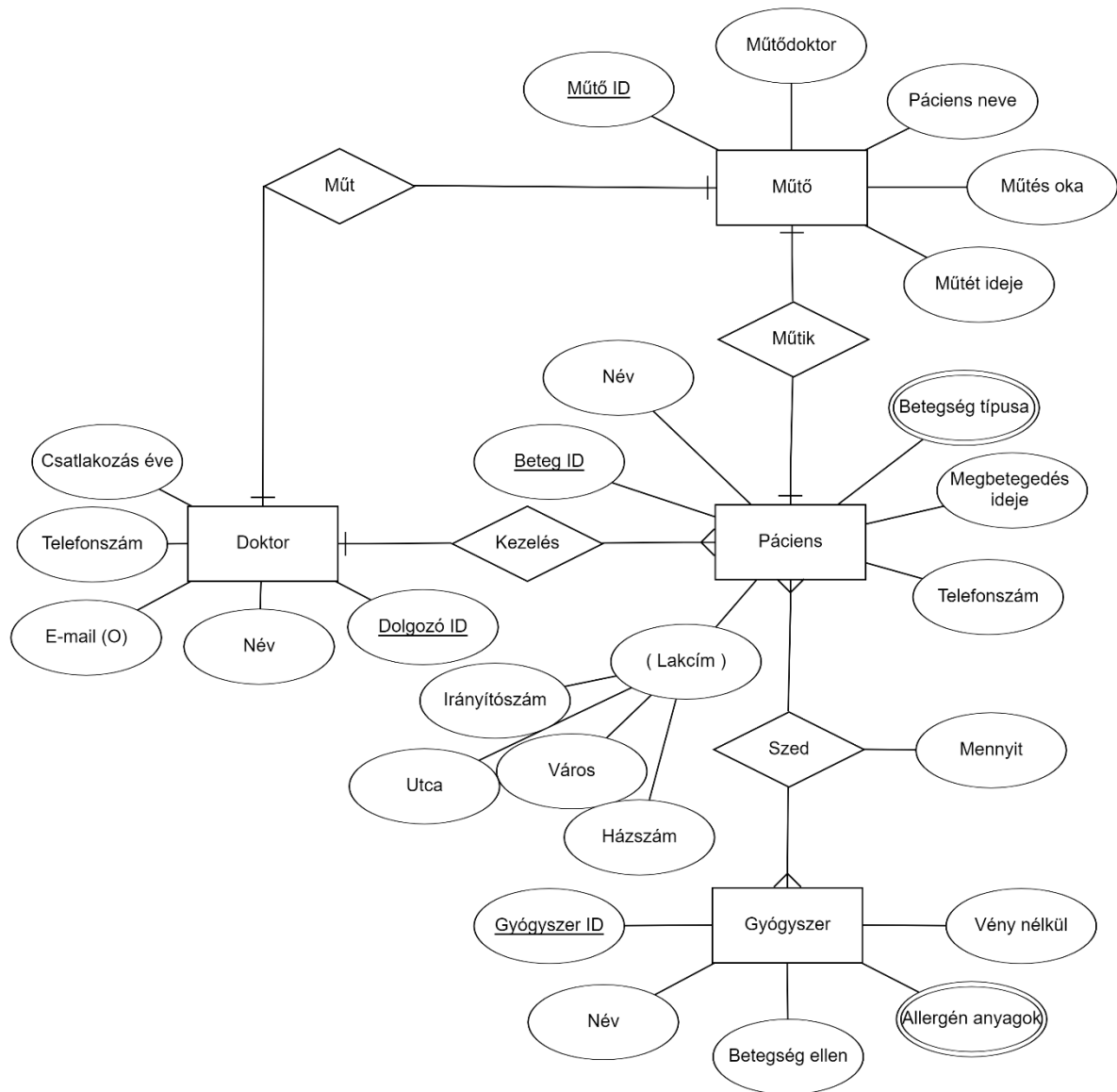
Elsődlegesen az adatbázisunkról egy ER modellt készítettem, majd azt egy XDM modellre konvertáltam.

Ez alapján elkészült az XML dokumentuma is már az adatbázisnak, melyhez egy sémát is létrehoztam.

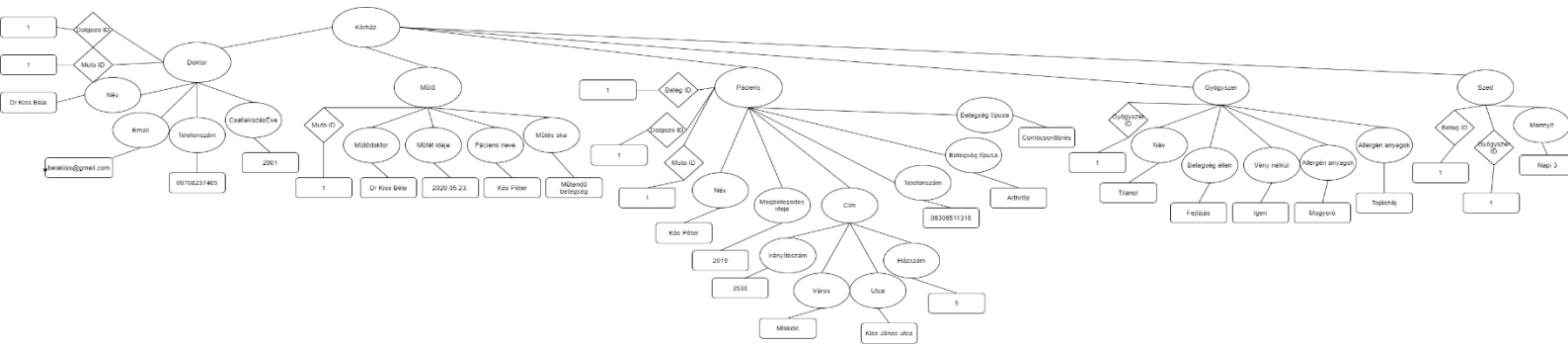
Továbbá egy DOM parsert is készítettem, hogy az XML-ünket olvasható formában kiírjuk bárkinek. Készült egy DOM modifier is, amellyel az egyes műtőkben tudjuk a műtés idejét módosítani.

1. feladat

1a) Az adatbázis ER modellje:



1b) Az adatbázis konvertálása XDM modellre:



1c) Az XDM modell alapján XML dokumentum készítése:

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

```
<korhaz>
```

```
  <muto mutoId="1">
    <mutodoktor>Dr Kiss Béla</mutodoktor>
    <mutetIdeje>2020.05.23</mutetIdeje>
    <paciensNeve>Kós Péter</paciensNeve>
    <mutesOka>Műtendő betegség</mutesOka>
  </muto>
```

```
  <muto mutoId="2">
    <mutodoktor>Dr János Pál</mutodoktor>
    <mutetIdeje>2020.09.13</mutetIdeje>
    <paciensNeve>Koaxk Ábel</paciensNeve>
    <mutesOka>Műtendő betegség</mutesOka>
  </muto>
```

```
  <doktor dolgozoId="1" mutoId="1">
    <nev>Dr Kiss Béla</nev>
    <telefonszam>06708237465</telefonszam>
    <csatlakozasEve>2001</csatlakozasEve>
    <email>belakiss@gmail.com</email>
  </doktor>
```

```
  <doktor dolgozoId="2" mutoId="2">
    <nev>Dr János Pál</nev>
    <telefonszam>06506327463</telefonszam>
    <csatlakozasEve>2002</csatlakozasEve>
  </doktor>
```

```
  <paciens betegId="1" dolgozoId="1" mutoId="1">
    <nev>Kós Péter</nev>
    <megbetegedesIdeje>2019</megbetegedesIdeje>
    <cim>
      <iranyitoszam>3530</iranyitoszam>
      <varos>Miskolc</varos>
      <utca>Kiss János utca</utca>
      <hazszam>5</hazszam>
    </cim>
    <telefonszam>06306511315</telefonszam>
    <betegsegTipusa>Arthritis</betegsegTipusa>
    <betegsegTipusa>Combcsonttörés</betegsegTipusa>
  </paciens>
```

```
  <paciens betegId="2" dolgozoId="2" mutoId="2">
    <nev>Koaxk Ábel</nev>
    <megbetegedesIdeje>2020</megbetegedesIdeje>
    <cim>
      <iranyitoszam>3531</iranyitoszam>
      <varos>Miskolc</varos>
      <utca>Péter Pál utca</utca>
      <hazszam>3</hazszam>
    </cim>
    <telefonszam>06307123272</telefonszam>
    <betegsegTipusa>Mogyoró allergia</betegsegTipusa>
    <betegsegTipusa>Tüdőembólia</betegsegTipusa>
  </paciens>
```

```

<gyogyszer gyógyszerId="1">
  <nev>Tilanol</nev>
  <betegsegEllen>Fejfájás</betegsegEllen>
  <venyNelkul>Igen</venyNelkul>
  <allergenAnyagok>Mogyoró</allergenAnyagok>
  <allergenAnyagok>Tojáshéj</allergenAnyagok>
</gyogyszer>

<szed betegId="1" gyógyszerId="1">
  <mennyit>Napi 3</mennyit>
</szed>

<szed betegId="2" gyógyszerId="1">
  <mennyit>Napi 10</mennyit>
</szed>

</korhaz>

```

1d) Az XML dokumentum alapján XMLSchema készítése:

```

<schema xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.example.org/XMLSchemaNRTHZZ"
xmlns:tns="http://www.example.org/XMLSchemaNRTHZZ" elementFormDefault="qualified">

  <element name="korhaz" type="tns:korhazTipus"/>

  <complexType name="mutoTipus">
    <sequence>
      <element name="mutoDoktor" type="string"/>
      <element name="mutetIdeje" type="string"/>
      <element name="paciensNeve" type="string"/>
      <element name="mutesOka" type="string"/>
    </sequence>
    <attribute name="mutoId" type="string" use="required"/>
  </complexType>

  <complexType name="doktorTipus">
    <sequence>
      <element name="nev" type="string"/>
      <element name="telefonszam" type="string"/>
      <element name="csatlakozasEve" type="integer"/>
      <element name="email" type="string" minOccurs="0"/>
    </sequence>
    <attribute name="dolgozoId" type="string" use="required"/>
    <attribute name="mutoId" type="IDREF"/>
  </complexType>

  <complexType name="cimTipus">
    <sequence>
      <element name="iranyitoszam" type="integer"/>
      <element name="varos" type="string"/>
      <element name="utca" type="string"/>
      <element name="hazszam" type="integer"/>
    </sequence>
  </complexType>

  <complexType name="paciensTipus">
    <sequence>
      <element name="nev" type="string"/>

```

```

        <element name="megbetegedesIdeje" type="integer"/>
        <element name="cim" type="tns:cimTipus"/>
        <element name="telefonszam" type="string"/>
        <element name="betegsegTipusa" type="string"
maxOccurs="unbounded"/>
    </sequence>
    <attribute name="betegId" type="string" use="required"/>
    <attribute name="dolgozoId" type="string" use="required"/>
    <attribute name="mutoId" type="IDREF"/>
</complexType>

<complexType name="gyogyszerTipus">
    <sequence>
        <element name="nev" type="string"/>
        <element name="betegsegEllen" type="string"/>
        <element name="venyNelkul" type="string"/>
        <element name="allergenAnyagok" type="string"
maxOccurs="unbounded"/>
    </sequence>
    <attribute name="gyogyszerId" type="string" use="required"/>
</complexType>

<complexType name="szedTipus">
    <sequence>
        <element name="mennyit" type="string"/>
    </sequence>
    <attribute name="betegId" type="IDREF"/>
    <attribute name="gyogyszerId" type="IDREF"/>
</complexType>

<complexType name="korhazTipus">
    <sequence>
        <element name="muto" type="tns:mutoTipus"
maxOccurs="unbounded"/>
        <element name="doktor" type="tns:doktorTipus"
maxOccurs="unbounded"/>
        <element name="paciens" type="tns:paciensTipus"
maxOccurs="unbounded"/>
        <element name="gyogyszer" type="tns:gyogyszerTipus"
maxOccurs="unbounded"/>
        <element name="szed" type="tns:szedTipus"
maxOccurs="unbounded"/>
    </sequence>
</complexType>

</schema>

```

2. feladat

2a) Az adatolvasás:

```

package hu.domparse.nrthzz;

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.TransformerException;

import java.io.File;
import java.io.IOException;

public class DOMReadNRTHZZ {

    public static void main(String args[]) throws ParserConfigurationException,
    IOException, SAXException, TransformerException {
        try {
            File inputFile = new File("XMLNRTHZZ.xml");
            DocumentBuilderFactory dbFactory =
            DocumentBuilderFactory.newInstance();

            DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
            Document doc = dBuilder.parse(inputFile);
            doc.getDocumentElement().normalize();
            System.out.println("Gyökér elem : " +
            doc.getDocumentElement().getNodeName());
            System.out.println("-----");

            NodeList nList = doc.getElementsByTagName("muto");
            for (int i = 0; i < nList.getLength(); i++) {
                Node nNode = nList.item(i);
                System.out.println("\nJelen elem: " +
            nNode.getNodeName());

                if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                    Element eElement = (Element) nNode;
                    System.out.println("Műtő id : " +
            eElement.getAttribute("mutoId"));
                    System.out.println("Műtődoktor: " +
            eElement.getElementsByTagName("mutodoktor").item(0).getTextContent());
                    System.out.println("Műtét ideje: " +
            eElement.getElementsByTagName("mutetIdeje").item(0).getTextContent());
                    System.out.println("Páciens neve: " +
            eElement.getElementsByTagName("paciensNeve").item(0).getTextContent());
                    System.out.println("Műtét oka: " +
            eElement.getElementsByTagName("mutesOka").item(0).getTextContent());
                }
            }

            nList = doc.getElementsByTagName("doktor");
            for (int i = 0; i < nList.getLength(); i++) {
                Node nNode = nList.item(i);
                System.out.println("\nJelen elem: " +
            nNode.getNodeName());

                if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                    Element eElement = (Element) nNode;
                    System.out.println("Doktor id : " +
            eElement.getAttribute("dolgozoId"));
                    System.out.println("Műtő id : " +
            eElement.getAttribute("mutoId"));
                    System.out.println("Név: " +
            eElement.getElementsByTagName("nev").item(0).getTextContent());
                    System.out.println("Telefonszám: " +
            eElement.getElementsByTagName("telefonszam").item(0).getTextContent());
                }
            }
        }
    }
}

```

```

        System.out.println("Csatlakozás éve: " +
eElement.getElementsByTagName("csatlakozasEve").item(0).getTextContent());
        //Opcionális, ezért amiatt, hogy ne dobjon null
        érték hibát, előre ellenőrizzük.
        //Ha nem null a mező, akkor kiírjuk
        //Ha null, akkor pedig jelezzük, hogy neki nincs
        ilyenje (jelen esetben e-mailje
        if(eElement.getElementsByTagName("email").item(0)
        != null){
            System.out.println("E-mail cím: "+
eElement.getElementsByTagName("email").item(0).getTextContent());
        }else {
            System.out.println("E-mail cím: Jelen
doktornak nincs e-mail címe!");
        }
    }

    //Páciensek kiírása
    nList = doc.getElementsByTagName("paciens");
    for (int i = 0; i < nList.getLength(); i++) {
        Node nNode = nList.item(i);
        System.out.println("\nJelen elem: " +
nNode.getNodeName());
        if (nNode.getNodeType() == Node.ELEMENT_NODE) {
            Element eElement = (Element) nNode;
            System.out.println("Beteg id : " +
eElement.getAttribute("betegId"));
            System.out.println("Doktor id : " +
eElement.getAttribute("dolgozoId"));
            System.out.println("Műtő id : " +
eElement.getAttribute("mutoId"));
            System.out.println("Név: " +
eElement.getElementsByTagName("nev").item(0).getTextContent());
            System.out.println("Megbetegedés ideje: " +
eElement.getElementsByTagName("megbetegedesIdeje").item(0).getTextContent());
            System.out.println("Beteg címe: ");
            System.out.println("Írányítószám: " +
eElement.getElementsByTagName("iranyitoszam").item(0).getTextContent());
            System.out.println("Város: " +
eElement.getElementsByTagName("varos").item(0).getTextContent());
            System.out.println("Utca: " +
eElement.getElementsByTagName("utca").item(0).getTextContent());
            System.out.println("Házzám: " +
eElement.getElementsByTagName("hazzsam").item(0).getTextContent());
            System.out.println("Telefonszám: " +
eElement.getElementsByTagName("telefonszam").item(0).getTextContent());
            int betegségTipusokSzama =
eElement.getElementsByTagName("betegsegTipusa").getLength();
            //Mivel több értéke is lehet a betegség típusának,
            ezért ki kell írtani egy for ciklussal az összeset
            for(int j = 0; j < betegségTipusokSzama; j++) {
                System.out.println("Betegség típusa: "+
eElement.getElementsByTagName("betegsegTipusa").item(j).getTextContent());
            }
        }
    }

    //Gyógyszerek kiírása

```



```

        nList = doc.getElementsByTagName("gyogyszer");
        for (int i = 0; i < nList.getLength(); i++) {
            Node nNode = nList.item(i);
            System.out.println("\nJelen elem: " +
nNode.getNodeName());
                if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                    Element eElement = (Element) nNode;
                    System.out.println("Gyógyszer Id : " +
eElement.getAttribute("gyogyszerId"));
                    System.out.println("Név: " +
eElement.getElementsByTagName("nev").item(0).getTextContent());
                    System.out.println("Betegség ellen: " +
eElement.getElementsByTagName("betegsegEllen").item(0).getTextContent());
                    System.out.println("Vény nélkül: " +
eElement.getElementsByTagName("venyNelkul").item(0).getTextContent());
                    int allergenAnyagokSzama =
doc.getElementsByTagName("allergenAnyagok").getLength();
                    //Mivel több értéke is lehet az allergén
anyagoknak, ezért ki kell írtani egy for ciklussal az összeset
                    for(int j = 0; j < allergenAnyagokSzama; j++) {
                        System.out.println("Allergén anyag: "+
eElement.getElementsByTagName("allergenAnyagok").item(j).getTextContent());
                    }
                }
            }

//Szed kapcsolótábla kiríratása
nList = doc.getElementsByTagName("szed");
for (int i = 0; i < nList.getLength(); i++) {
    Node nNode = nList.item(i);
    System.out.println("\nJelen elem: " +
nNode.getNodeName());
        if (nNode.getNodeType() == Node.ELEMENT_NODE) {
            Element eElement = (Element) nNode;
            System.out.println("Beteg id : " +
eElement.getAttribute("betegId"));
            System.out.println("Gyógyszer id : " +
eElement.getAttribute("gyogyszerId"));
            System.out.println("Mennyit kell szednie? " +
eElement.getElementsByTagName("mennyit").item(0).getTextContent());
        }
    }
} catch (Exception e) {
    e.printStackTrace();
}
}
}

```

2b) Adatmódosítás:

```

package hu.domparsenrthzz;

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

        String filePath = "XMLNRTHZZ.xml";
        File xmlFile = new File(filePath);

        DocumentBuilderFactory dbFactory =
DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder;

        try {
            dBuilder = dbFactory.newDocumentBuilder();
            //XML betöltése és parseolása
            Document doc = dBuilder.parse(xmlFile);

            doc.getDocumentElement().normalize();

            //Elem módosítása
            mutetIdejenekModositasa(doc);

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

    //Függvény a módosítandó műtő idjének beolvasására
    public static String idBeolvasas() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Műtő id:");
        String id = sc.nextLine();
        return id;
    }

    //Függvény, amely létrehozza az új filet
    public static void xmlFileIras(Document doc) throws TransformerException {
        TransformerFactory transformerFactory = TransformerFactory.newInstance();
        Transformer transformer = transformerFactory.newTransformer();
        DOMSource source = new DOMSource(doc);
        StreamResult result = new StreamResult(new
File("src/hu/domparse/nrthzz/XMLNRTHZZ.updated.xml"));
        transformer.transform(source, result);
    }
}

```

```

        //Függvény amely megteszi a módosítást a felhasználó által megadott
        időpontra
        //Meghívja az xmlFileIras függvényt, amellyel létrehoz egy új fájlt
        public static void mutetIdejenekModositasa(Document doc) throws
        TransformerException {

            //Először beolvassuk az ID-t
            System.out.println("Melyik műtőben változik a műtész ideje?");
            String beolvasottId = idBeolvasas();

            //Majd a műtész új idejét
            Scanner sc = new Scanner(System.in);
            System.out.print("Műtész új ideje: ");
            String mutetIdeje = sc.nextLine();

            //Megkeressük azt a nodeot, aminek az ID-je egyezik a user által
            megadott idvel
            NodeList nList = doc.getElementsByTagName("muto");
            for (int i = 0; i < nList.getLength(); i++) {

                Node nNode = nList.item(i);

                if (nNode.getNodeType() == Node.ELEMENT_NODE) {

                    Element element = (Element) nNode;

                    String mutoId = element.getAttribute("mutoId");

                    if (mutoId.equals(beolvasottId)) {

                        Node node1 =
                        element.getElementsByTagName("mutetIdeje").item(0);
                        node1.setTextContent(mutetIdeje);

                        System.out.println("Sikeres Módosítás");
                        //Megírjuk az új XML filet
                        xmlFileIras(doc);
                    }
                }
            }
        }
    }
}

```