XML

Eniko Vadon

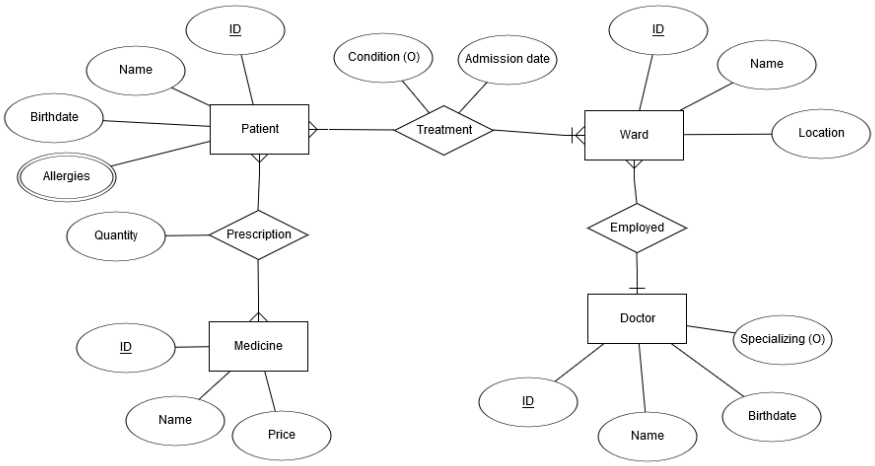
December 2020

1 Feladat

1.1 Leiras:

A kezelt adatbazisom egy korhazi nyilvantartas, amiben kulonbozo korhazi egysegeket, az ott dolgozo orvosokat es a bentfekvo betegeket tartja nyilvan, ezen kivul megjelenik benne a betegeknek adott kezelesek es a hozza tartozo gyogyszerek adatai is.

1.2 ER model



1.3 XML

```
<?xml version="1.0" encoding="utf-8"?>
<hospital xmlns:xsi="https://w3.org/2001/XMLSchema-instance">

        <ward id="wardccu">
                <name>CCU</name>
                <location>North Wing</location>
        </ward>
        <ward id="wardrehab">
                <name>Rehabilitation</name>
                <location>Southwest Wing</location>
        </ward>

        <patient id="pat23">
                <fullname>
                        <firstname>John</firstname>
```

```
                <lastname>Wick</lastname>
        </fullname>
        <birthdate>1993-03-02</birthdate>
        <allergy>Lactose</allergy>
        <allergy>penicillin</allergy>
</patient>
<patient id="pat92">
        <fullname>
                <firstname>Mike</firstname>
                <lastname>Ritch</lastname>
        </fullname>
        <birthdate>1978-06-30</birthdate>
        <allergy>Gluten</allergy>
</patient>

<doctor id="doc6">
        <fullname>
                <firstname>Charles</firstname>
                <lastname>Schultz</lastname>
        </fullname>
        <birthdate>1965-11-11</birthdate>
        <specializing>Pediatrics</specializing>
        <employed>wardccu</employed>
</doctor>
<doctor id="doc21">
        <fullname>
                <firstname>Celestina</firstname>
                <lastname>Warbeck</lastname>
        </fullname>
        <birthdate>1970.01.01</birthdate>
        <employed>wardccu</employed>
</doctor>

<medicine id="medpain1">
        <name>Aspirin</name>
        <price>2000</price>
</medicine>
<medicine id="medlax5">
        <name>Ducolax</name>
        <price>300</price>
</medicine>
<medicine id="medanx4">
        <name>Xanax</name>
        <price>666</price>
</medicine>
```

```
        <treatment>
                <patient-id>pat23</patient-id>
                <ward-id>wardccu</ward-id>
                <condition>arrhythmia</condition>
                <admission-date>2020-06-08</admission-date>
        </treatment>
        <treatment>
                <patient-id>pat92</patient-id>
                <ward-id>wardrehab</ward-id>
                <admission-date>2020-10-13</admission-date>
        </treatment>

        <prescription>
                <patient-id>pat92</patient-id>
                <medicine-id>medlax5</medicine-id>
                <quantity>30</quantity>
        </prescription>
        <prescription>
                <patient-id>pat92</patient-id>
                <medicine-id>medanx4</medicine-id>
                <quantity>20</quantity>
        </prescription>
        <prescription>
                <patient-id>pat23</patient-id>
                <medicine-id>medanx4</medicine-id>
                <quantity>60</quantity>
        </prescription>
</hospital>
```

1.4 XSD

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

        <xs:element name="location" type="xs:string" />
        <xs:element name="name" type="xs:string" />
        <xs:element name="firstname" type="xs:string" />
        <xs:element name="lastname" type="xs:string" />
        <xs:element name="birthdate" type="xs:string" />
        <xs:element name="allergy" type="xs:string" />
        <xs:element name="price" type="xs:short" />
        <xs:element name="quantity" type="xs:short" />
        <xs:element name="condition" type="xs:string" />
        <xs:element name="admission-date" type="xs:date" />
        <xs:element name="specializing" type="xs:string" />
```

```
<xs:element name="employed" type="xs:IDREF" />
<xs:element name="patient-id" type="xs:IDREF" />
<xs:element name="ward-id" type="xs:IDREF" />
<xs:element name="medicine-id" type="xs:IDREF" />


<xs:element name="hospital">
        <xs:complexType>
                <xs:sequence>
                        <xs:element ref="ward" maxOccurs="unbounded" />
                        <xs:element ref="patient" maxOccurs="unbounded"
                        <xs:element ref="doctor" maxOccurs="unbounded" /
                        <xs:element ref="medicine" maxOccurs="unbounded"
                        <xs:element ref="treatment" maxOccurs="unbounded
                        <xs:element ref="prescription" maxOccurs="unboun
                </xs:sequence>
        </xs:complexType>
</xs:element>

<xs:element name="ward">
        <xs:complexType mixed="true">
                <xs:sequence>
                        <xs:element ref="name" />
                        <xs:element ref="location" />
                </xs:sequence>
                <xs:attribute type="xs:ID" name="id" use="required" />
        </xs:complexType>
</xs:element>


<xs:element name="patient">
        <xs:complexType mixed="true">
                <xs:sequence>
                        <xs:element ref="fullname" />
                        <xs:element ref="birthdate" />
                        <xs:element ref="allergy" maxOccurs="unbounded"
                </xs:sequence>
                <xs:attribute type="xs:ID" name="id" use="required" />
        </xs:complexType>
</xs:element>

<xs:element name="fullname">
        <xs:complexType mixed="true">
                <xs:sequence>
                        <xs:element ref="firstname" minOccurs="1" />
```

```
                        <xs:element ref="lastname" minOccurs="1" />
                </xs:sequence>
        </xs:complexType>
</xs:element>



<xs:element name="doctor">
        <xs:complexType>
                <xs:sequence>
                        <xs:element ref="fullname" />
                        <xs:element ref="birthdate" />
                        <xs:element ref="specializing" minOccurs="0" />
                        <xs:element ref="employed" />
                </xs:sequence>
                <xs:attribute type="xs:ID" name="id" use="required" />
        </xs:complexType>
</xs:element>

<xs:element name="medicine">
        <xs:complexType mixed="true">
                <xs:sequence>
                        <xs:element ref="name" />
                        <xs:element ref="price" />
                </xs:sequence>
                <xs:attribute type="xs:ID" name="id" use="required" />
        </xs:complexType>
</xs:element>

<xs:element name="treatment">
        <xs:complexType>
                <xs:sequence>
                        <xs:element ref="patient-id" />
                        <xs:element ref="ward-id" />
                        <xs:element ref="condition" minOccurs="0" />
                        <xs:element ref="admission-date" />
                </xs:sequence>
        </xs:complexType>
</xs:element>

<xs:element name="prescription">
        <xs:complexType>
                <xs:sequence>
                        <xs:element ref="patient-id" />
                        <xs:element ref="medicine-id" />
                        <xs:element ref="quantity" />
```

```
                    </xs:sequence>
                </xs:complexType>
        </xs:element>

</xs:schema>
```

2 Kod

2.1 Adatolvasas - DOMDataReader.java

```
package domData;

import java.io.File;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;

import org.w3c.dom.DOMException;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;

public class DOMDataReader {

        public static void main(String[] args) {
                try {
                        DocumentBuilderFactory documentBuilderFactory = Documen
                        DocumentBuilder documentBuilder = documentBuilderFactory
                        File file = new File("hospital.xml");
                        Document doc = documentBuilder.parse(file);

                        doc.getDocumentElement().normalize();
                        Element rootElement = doc.getDocumentElement();

                        System.out.println(rootElement.getNodeName());

                        // readWards(rootElement);
                        // readDoctors(rootElement);
                        // readPatients(rootElement);
                        // readMedicines(rootElement);
                        // readTreatments(rootElement);

                } catch (Exception E) {
```

```
                E.printStackTrace();
        }

}

public static void readWards(Element rootElement) {
        try {
                NodeList nodeList = rootElement.getElementsByTagName("wa

                for (int i = 0; i < nodeList.getLength(); i++) {
                        Node currentNode = nodeList.item(i);
                        if (currentNode.getNodeType() == Node.ELEMENT_NO
                                Element currentElement = (Element) curre

                                String id = currentElement.getAttribute(
                                System.out.println("Id: " + id);

                                String name = currentElement.getElementsB
                                System.out.println("\tName: " + name);

                                String location = currentElement.getEleme
                                System.out.println("\tLocation: " + loca
                        }
                }
        } catch (DOMException e) {
                e.printStackTrace();
        }
}

public static void readDoctors(Element rootElement) {
        try {
                NodeList nodeList = rootElement.getElementsByTagName("do

                for (int i = 0; i < nodeList.getLength(); i++) {
                        Node currentNode = nodeList.item(i);
                        if (currentNode.getNodeType() == Node.ELEMENT_NO
                                Element currentElement = (Element) curre

                                String id = currentElement.getAttribute(
                                System.out.println("id: " + id);

                                String fullname = currentElement.getEleme
                                System.out.println("\tName: " + fullname

                                String birthdate = currentElement.getElem
                                System.out.println("\tBirthdate: " + bir
```

```
                                        // optional node
                                        NodeList specializingNode = currentEleme
                                        for (int j = 0; j < specializingNode.get
                                                if (specializingNode.item(j).getN
                                                        String specializing = sp
                                                        System.out.println("\tSp
                                                }
                                        }

                                        // id ref
                                        String employed = currentElement.getEleme
                                        NodeList wardNodes = rootElement.getEleme
                                        for (int k = 0; k < nodeList.getLength()
                                                Node currentWardNode = wardNodes
                                                if (currentWardNode.getNodeType(
                                                        Element currentWardEleme
                                                        if (currentWardElement.g
                                                                String wardName =

                                                                System.out.print
                                                        }
                                                }
                                        }
                                }
                        }
                } catch (DOMException e) {
                        e.printStackTrace();
                }
        }

        public static void readPatients(Element rootElement) {
                try {
                        NodeList nodeList = rootElement.getElementsByTagName("pa

                        for (int i = 0; i < nodeList.getLength(); i++) {
                                Node currentNode = nodeList.item(i);
                                if (currentNode.getNodeType() == Node.ELEMENT_NO
                                        Element currentElement = (Element) curren

                                        String id = currentElement.getAttribute(
                                        System.out.println("id: " + id);

                                        String fullname = currentElement.getEleme
                                        System.out.println("\tName: " + fullname
```

```
                                String birthdate = currentElement.getElem
                                System.out.println("\tBirthdate: " + bir

                                // multivalued
                                for (int j = 0; j < currentElement.getEle
                                        String allergy = currentElement.
                                        System.out.println("\tAllergy: "
                                }

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

}

public static void readMedicines(Element rootElement) {
        try {
                NodeList nodeList = rootElement.getElementsByTagName("me

                for (int i = 0; i < nodeList.getLength(); i++) {
                        Node currentNode = nodeList.item(i);
                        if (currentNode.getNodeType() == Node.ELEMENT_NO
                                Element currentElement = (Element) curre

                                String id = currentElement.getAttribute(
                                System.out.println("id: " + id);

                                String name = currentElement.getElementsB
                                System.out.println("\tName: " + name);

                                String price = currentElement.getElements
                                System.out.println("\tPrice: " + price);

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

public static void readTreatments(Element rootElement) {

        NodeList nodeList = rootElement.getElementsByTagName("treatment"
```

```
                for (int i = 0; i < nodeList.getLength(); i++) {
                        Node currentNode = nodeList.item(i);
                        if (currentNode.getNodeType() == Node.ELEMENT_NODE) {
                                Element currentElement = (Element) currentNode;

                                //patient
                                String patient = currentElement.getElementsByTagN
                                NodeList patientNodes = rootElement.getElementsBy
                                for (int j = 0; j < patientNodes.getLength(); j+
                                        Node currentPatientNode = patientNodes.it
                                        if (currentPatientNode.getNodeType() ==
                                                Element currentPatientElement =
                                                if (currentPatientElement.getAtt
                                                        String patientName = cur
                                                        System.out.println("\tPa
                                                }
                                        }
                                }

                                // id ref
                                String ward = currentElement.getElementsByTagNam
                                NodeList wardNodes = rootElement.getElementsByTag
                                for (int j = 0; j < nodeList.getLength(); j++) {
                                        Node currentWardNode = wardNodes.item(j)
                                        if (currentWardNode.getNodeType() == Nod
                                                Element currentWardElement = (El
                                                if (currentWardElement.getAttrib
                                                        String wardName = current
                                                                        .getText
                                                        System.out.println("\tTr
                                                }
                                        }
                                }
                        }
                }
        }
}
```

2.2 Adatmodositas - DOMDataModifier.java

```
package domData;

import java.io.File;

import javax.xml.parsers.DocumentBuilder;
```

```
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;

import org.w3c.dom.DOMException;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.NamedNodeMap;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;

public class DOMDataModifier {
        public static void main(String[] args) {

                try {
                        File inputFile = new File("hospital.xml");
                        DocumentBuilderFactory docFactory = DocumentBuilderFacto
                        DocumentBuilder docBuilder = docFactory.newDocumentBuild
                        Document doc = docBuilder.parse(inputFile);
                        Element rootElement = doc.getDocumentElement();

                        String oldPatientID = "pat69", newPatientID = "pat11";
                        modifyPatientID(rootElement, oldPatientID, newPatientID)
                        String medicineName = "Aspirin";
                        int newPrice = 1662;
                        updateMedicinePrice(rootElement, medicineName, newPrice)

                        // write the content on console
                        TransformerFactory transformerFactory = TransformerFacto
                        Transformer transformer = transformerFactory.newTransfor
                        DOMSource source = new DOMSource(doc);
                        StreamResult outputResult = new StreamResult(inputFile);
                        transformer.transform(source, outputResult);

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

        public static void modifyPatientID(Element rootElement, String oldPatient
                try {
                        NodeList patientList = rootElement.getElementsByTagName(
                        // update patient id
                        for (int i = 0; i < patientList.getLength(); i++) {
```

```
                                        NamedNodeMap attr = patientList.item(i).getAttri
                                        Node nodeAttr = attr.getNamedItem("id");
                                        if (oldPatientID.equalsIgnoreCase(nodeAttr.getTe
                                                nodeAttr.setTextContent(newPatientID);
                                                System.out.println("Modified patient wit
                                        }
                                }
                        } catch (DOMException e) {
                                e.printStackTrace();
                        }
                }

                public static void updateMedicinePrice(Element rootElement, String medici
                        try {
//                              all medicine nodes
                                NodeList medicineList = rootElement.getElementsByTagName
//                              all nodes of one medicine
                                NodeList allMedicine = medicineList.item(0).getChildNode
                                for (int i = 0; i < medicineList.getLength(); i++) {
//                                      test if current node is the same as the one we a
                                        Node medNode = allMedicine.item(i);
                                        if (medNode.getNodeName().equals("name") && medi
                                                Element medElement = (Element) medNode.g
//                                                      System.out.println(medElement.get
                                                medElement.getElementsByTagName("price")
                                                System.out.println("Modified price of "

                                                break;
                                        }
                                }
                        } catch (DOMException e) {
                                e.printStackTrace();
                        }

                }
        }
```