XML

Eniko Vadon

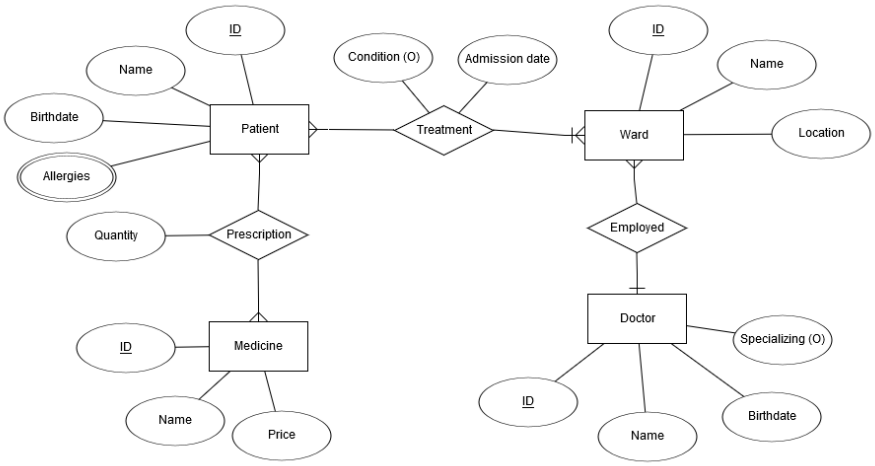
December 2020

1 Feladat

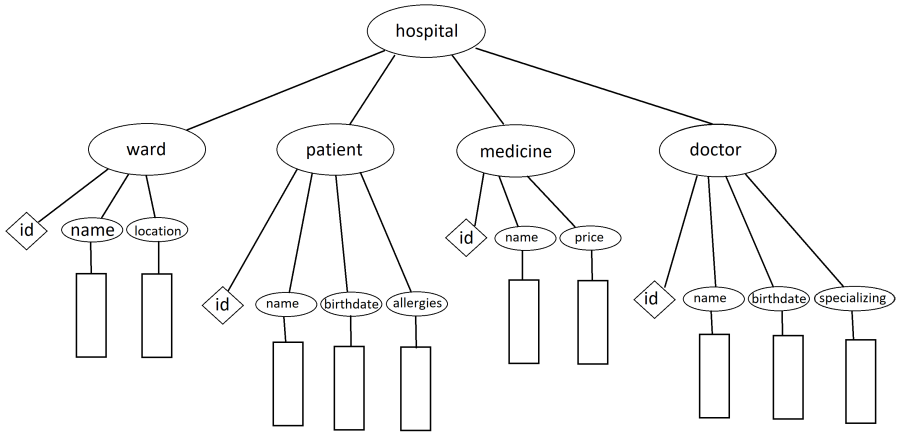
1.1 Leiras:

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 XDM model



1.4 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>
```

1.5 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

DOMDataReader.java | DOMDataModifier.java | hospital.xml | hospital.xsd

```java
package domData;

import java.io.File;

public class DOMDataReader {

    public static void main(String[] args) {
        try {
            DocumentBuilderFactory documentBuilderFactory = DocumentBuilderFactory.newInstance();
            DocumentBuilder documentBuilder = documentBuilderFactory.newDocumentBuilder();
            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);
            readDocument(rootElement);

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

    }

    public static void readDocument (Element rootElement) {
        readWards(rootElement);
        readDoctors(rootElement);
        readPatients(rootElement);
        readMedicines(rootElement);
        readTreatments(rootElement);
    }
```

DOMDataReader.java | DOMDataModifier.java | hospital.xml | hospital.xsd

```java
    public static void readWards(Element rootElement) {
        System.out.println("\nWards: ");
        try {
            NodeList nodeList = rootElement.getElementsByTagName("ward");

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

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

                    String name = currentElement.getElementsByTagName("name").item(0).getTextContent();
                    System.out.println("\tName: " + name);

                    String location = currentElement.getElementsByTagName("location").item(0).getTextContent();
                    System.out.println("\tLocation: " + location);
                }
            }
        } catch (DOMException e) {
            e.printStackTrace();
        }
    }

    public static void readDoctors(Element rootElement) {
        System.out.println("\nDoctors: ");
        try {
            NodeList nodeList = rootElement.getElementsByTagName("doctor");

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

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

                    String fullname = currentElement.getElementsByTagName("fullname").item(0).getTextContent();
                    System.out.println("\tName: " + fullname);

                    String birthdate = currentElement.getElementsByTagName("birthdate").item(0).getTextContent();
```

DOMDataReader.java | DOMDataModifier.java | hospital.xml | hospital.xsd

```java
        }
    }

    public static void readDoctors(Element rootElement) {
        System.out.println("\nDoctors: ");
        try {
            NodeList nodeList = rootElement.getElementsByTagName("doctor");

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

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

                    String fullname = currentElement.getElementsByTagName("fullname").item(0).getTextContent();
                    System.out.println("\tName: " + fullname);

                    String birthdate = currentElement.getElementsByTagName("birthdate").item(0).getTextContent();
                    System.out.println("\tBirthdate: " + birthdate);

                    // optional node
                    NodeList specializingNode = currentElement.getChildNodes();
                    for (int j = 0; j < specializingNode.getLength(); j++) {
                        if (specializingNode.item(j).getNodeName() == "specializing") {
                            String specializing = specializingNode.item(j).getTextContent();
                            System.out.println("\tSpecializing: " + specializing);
                        }
                    }

                    // id ref
                    String employed = currentElement.getElementsByTagName("employed").item(0).getTextContent();
                    NodeList wardNodes = rootElement.getElementsByTagName("ward");
                    for (int k = 0; k < nodeList.getLength(); k++) {
                        Node currentWardNode = wardNodes.item(k);
                        if (currentWardNode.getNodeType() == Node.ELEMENT_NODE) {
                            Element currentWardElement = (Element) currentWardNode;
                            if (currentWardElement.getAttribute("id").compareToIgnoreCase(employed) == 0) {
                                String wardName = currentWardElement.getElementsByTagName("name").item(0)
                                        .getTextContent();
                                System.out.println("\tEmployed at: " + wardName);
                            }
```

DOMDataReader.java | DOMDataModifier.java | hospital.xml | hospital.xsd

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

    public static void readPatients(Element rootElement) {
        System.out.println("\nPatients");
        try {
            NodeList nodeList = rootElement.getElementsByTagName("patient");

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

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

//                  NodeList nameNodes=currentElement.getElementsByTagName("fullname").item(0).getChildNodes();
//                  String fullname="";
//                  for (int j = 0; j < nameNodes.getLength(); j++) {
//                      fullname+=nameNodes.item(j).getTextContent();
//                  }
                    String fullname = currentElement.getElementsByTagName("fullname").item(0).getTextContent();
                    System.out.println("\tName: " + fullname);

                    String birthdate = currentElement.getElementsByTagName("birthdate").item(0).getTextContent();
                    System.out.println("\tBirthdate: " + birthdate);

                    // multivalued
                    for (int j = 0; j < currentElement.getElementsByTagName("allergy").getLength(); j++) {
                        String allergy = currentElement.getElementsByTagName("allergy").item(j).getTextContent();
                        System.out.println("\tAllergy: " + allergy);
                    }

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

```java
    public static void readMedicines(Element rootElement) {
        System.out.println("\nMedicines: ");
        try {
            NodeList nodeList = rootElement.getElementsByTagName("medicine");

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

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

                    String name = currentElement.getElementsByTagName("name").item(0).getTextContent();
                    System.out.println("\tName: " + name);

                    String price = currentElement.getElementsByTagName("price").item(0).getTextContent();
                    System.out.println("\tPrice: " + price);

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

    public static void readTreatments(Element rootElement) {
        System.out.println("\nTreatments: ");
        try {
            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.getElementsByTagName("patient-id").item(0).getTextContent();
                    NodeList patientNodes = rootElement.getElementsByTagName("patient");
                    for (int j = 0; j < patientNodes.getLength(); j++) {
                        Node currentPatientNode = patientNodes.item(j);
                        if (currentPatientNode.getNodeType() == Node.ELEMENT_NODE) {
                            Element currentPatientElement = (Element) currentPatientNode;
```

2.2 Adatmodositas - DOMDataModifier.java

DOMDataReader.java | DOMDataModifier.java | hospital.xml | hospital.xsd

```java
package domData;

import java.io.File;

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

        try {
            File inputFile = new File("hospital.xml");
            DocumentBuilderFactory docFactory = DocumentBuilderFactory.newInstance();
            DocumentBuilder docBuilder = docFactory.newDocumentBuilder();
            Document doc = docBuilder.parse(inputFile);

            Element rootElement = doc.getDocumentElement();

            String oldPatientID = "pat23", newPatientID = "pat11";
            modifyPatientID(rootElement, oldPatientID, newPatientID);
            String medicineName = "Aspirin";
            int newPrice = 1662;
            updateMedicinePrice(rootElement, medicineName, newPrice);

            // write the content on back into the file
            TransformerFactory transformerFactory = TransformerFactory.newInstance();
            Transformer transformer = transformerFactory.newTransformer();
            DOMSource source = new DOMSource(doc);
            StreamResult outputResult = new StreamResult(inputFile);
            transformer.transform(source, outputResult);

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

DOMDataReader.java | DOMDataModifier.java | hospital.xml | hospital.xsd

```java
    public static void modifyPatientID(Element rootElement, String oldPatientID, String newPatientID) {
        try {
            NodeList patientList = rootElement.getElementsByTagName("patient");
            // update patient id
            for (int i = 0; i < patientList.getLength(); i++) {
                NamedNodeMap attr = patientList.item(i).getAttributes();
                Node nodeAttr = attr.getNamedItem("id");
                if (oldPatientID.equalsIgnoreCase(nodeAttr.getTextContent())) {
                    nodeAttr.setTextContent(newPatientID);
                    System.out.println("Modified patient with id " + oldPatientID + ".");
                    break;
                }
            }
        } catch (DOMException e) {
            e.printStackTrace();
        }
    }

    public static void updateMedicinePrice(Element rootElement, String medicineName, int newPrice) {
        try {
//          all medicine nodes
            NodeList medicineList = rootElement.getElementsByTagName("medicine");
//          all nodes of one medicine
            NodeList allMedicine = medicineList.item(0).getChildNodes();
            for (int i = 0; i < medicineList.getLength(); i++) {
//              test if current node is the same as the one we are looking for
                Node medNode = allMedicine.item(i);
                if (medNode.getNodeName().equals("name") && medicineName.equalsIgnoreCase(medNode.getTextContent())) {
                    Element medElement = (Element) medNode.getParentNode();
//                  System.out.println(medElement.getElementsByTagName("price").item(0).getTextContent());
                    medElement.getElementsByTagName("price").item(0).setTextContent(String.valueOf(newPrice));
                    System.out.println("Modified price of " + medicineName + " to " + newPrice);
                    break;
                }
            }
        } catch (DOMException e) {
            e.printStackTrace();
        }

    }
}
```