

---

# Jegyzokonyv

---

ADATKEZELES XML KORNYEZETBEN

FELEVES FELADAT

KESZITETTE: VADON ENIKO

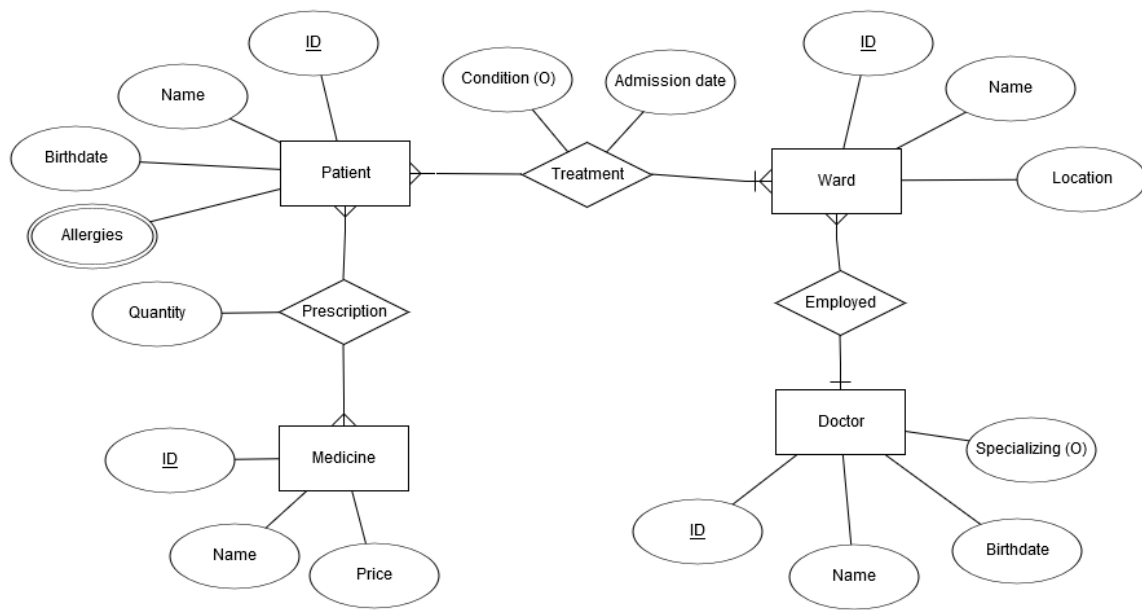
NEPTUNKOD: DW9YG6

# 1 Feladat

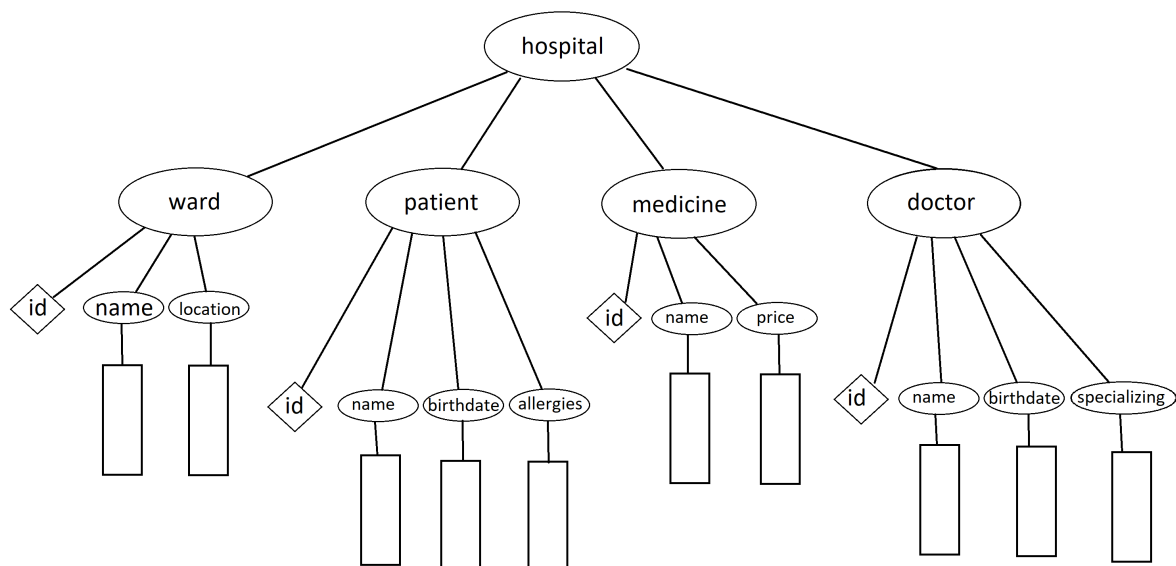
## 1.1 Leírás:

A kezelt adatbázisom egy kórházi nyilvántartás, amiben különbozo kórházi egysegeket, az ott dolgozo orvosokat es a bentfekvo betegeket tartja nyilvan, ezen kivul megjelenik benne a betegeknak 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>
</hospital>
```

```

</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="unbounded" />
    </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
14 public class DOMDataReader {
15
16     public static void main(String[] args) {
17         try {
18             //Initialize a DocumentBuilderFactory
19             //Dokumentumolvaso letrehozasa
20             DocumentBuilderFactory documentBuilderFactory = DocumentBuilderFactory.newInstance();
21             DocumentBuilder documentBuilder = documentBuilderFactory.newDocumentBuilder();
22             //Read the XML file
23             //XML Dokumentum beolvasasa
24             File file = new File("hospital.xml");
25             //Parse the document into readable format
26             //A letezo dokumentum feldolgozasa a dokumentumolvasoval
27             Document doc = documentBuilder.parse(file);
28             //Normalize
29             //Dokumentum normalizalasa
30             doc.getDocumentElement().normalize();
31             //Get the rootelement
32             //Gyokerelem elerese
33             Element rootElement = doc.getDocumentElement();
34
35             System.out.println(rootElement.getNodeName());
36
37             //      readWards(rootElement);
38             //      readDoctors(rootElement);
39             //      readPatients(rootElement);
40             //      readMedicines(rootElement);
41             //      readTreatments(rootElement);
42             readDocument(rootElement);
43         }
44     }
45 }

DOMDataReader.java DOMDataModifier.java hospital.xml hospital.xsd
48
49     public static void readWards(Element rootElement) {
50         System.out.println("\nWards: ");
51         try {
52             NodeList nodeList = rootElement.getElementsByTagName("ward");
53
54             for (int i = 0; i < nodeList.getLength(); i++) {
55                 Node currentNode = nodeList.item(i);
56                 if (currentNode.getNodeType() == Node.ELEMENT_NODE) {
57                     Element currentElement = (Element) currentNode;
58
59                     String id = currentElement.getAttribute("id");
60                     System.out.println("Id: " + id);
61
62                     String name = currentElement.getElementsByTagName("name").item(0).getTextContent();
63                     System.out.println("\tName: " + name);
64
65                     String location = currentElement.getElementsByTagName("location").item(0).getTextContent();
66                     System.out.println("\tLocation: " + location);
67                 }
68             }
69         } catch (DOMException e) {
70             e.printStackTrace();
71         }
72     }
73
74     public static void readDoctors(Element rootElement) {
75         System.out.println("\nDoctors: ");
76         try {
77             NodeList nodeList = rootElement.getElementsByTagName("doctor");
78
79             for (int i = 0; i < nodeList.getLength(); i++) {
80                 Node currentNode = nodeList.item(i);
81                 if (currentNode.getNodeType() == Node.ELEMENT_NODE) {
82                     Element currentElement = (Element) currentNode;
83
84                     String id = currentElement.getAttribute("id");
85                     System.out.println("id: " + id);
86
87                     String fullname = currentElement.getElementsByTagName("fullname").item(0).getTextContent();
88                     System.out.println("\tName: " + fullname);
89
90                     String birthdate = currentElement.getElementsByTagName("birthdate").item(0).getTextContent();
```

```

DOMDataReader.java DOMDataModifier.java hospital.xml hospital.xsd
71 }
72 }
73
74 public static void readDoctors(Element rootElement) {
75     System.out.println("\nDoctors: ");
76     try {
77         NodeList nodeList = rootElement.getElementsByTagName("doctor");
78
79         for (int i = 0; i < nodeList.getLength(); i++) {
80             Node currentNode = nodeList.item(i);
81             if (currentNode.getNodeType() == Node.ELEMENT_NODE) {
82                 Element currentElement = (Element) currentNode;
83
84                 String id = currentElement.getAttribute("id");
85                 System.out.println("id: " + id);
86
87                 String fullname = currentElement.getElementsByTagName("fullname").item(0).getTextContent();
88                 System.out.println("\tName: " + fullname);
89
90                 String birthdate = currentElement.getElementsByTagName("birthdate").item(0).getTextContent();
91                 System.out.println("\tBirthdate: " + birthdate);
92
93                 // optional node
94                 NodeList specializingNode = currentElement.getChildNodes();
95                 for (int j = 0; j < specializingNode.getLength(); j++) {
96                     if (specializingNode.item(j).getNodeName() == "specializing") {
97                         String specializing = specializingNode.item(j).getTextContent();
98                         System.out.println("\tSpecializing: " + specializing);
99                     }
100                 }
101
102                 // id ref
103                 String employed = currentElement.getElementsByTagName("employed").item(0).getTextContent();
104                 NodeList wardNodes = rootElement.getElementsByTagName("ward");
105                 for (int k = 0; k < wardNodes.getLength(); k++) {
106                     Node currentWardNode = wardNodes.item(k);
107                     if (currentWardNode.getNodeType() == Node.ELEMENT_NODE) {
108                         Element currentWardElement = (Element) currentWardNode;
109                         if (currentWardElement.getAttribute("id").compareToIgnoreCase(employed) == 0) {
110                             String wardName = currentWardElement.getElementsByTagName("name").item(0)
111                                 .getTextContent();
112                             System.out.println("\tEmployed at: " + wardName);
113                         }
114                     }
115                 }
116             }
117         }
118     } catch (DOMException e) {
119         e.printStackTrace();
120     }
121 }
122
123 public static void readPatients(Element rootElement) {
124     System.out.println("\nPatients");
125     try {
126         NodeList nodeList = rootElement.getElementsByTagName("patient");
127
128         for (int i = 0; i < nodeList.getLength(); i++) {
129             Node currentNode = nodeList.item(i);
130             if (currentNode.getNodeType() == Node.ELEMENT_NODE) {
131                 Element currentElement = (Element) currentNode;
132
133                 String id = currentElement.getAttribute("id");
134                 System.out.println("id: " + id);
135
136                 //
137                 //
138                 //
139                 //
140                 //
141                 String fullname = currentElement.getElementsByTagName("fullname").item(0).getTextContent();
142                 System.out.println("\tName: " + fullname);
143
144                 String birthdate = currentElement.getElementsByTagName("birthdate").item(0).getTextContent();
145                 System.out.println("\tBirthdate: " + birthdate);
146
147                 // multivalued
148                 for (int j = 0; j < currentElement.getElementsByTagName("allergy").getLength(); j++) {
149                     String allergy = currentElement.getElementsByTagName("allergy").item(j).getTextContent();
150                     System.out.println("\tAllergy: " + allergy);
151                 }
152             }
153         }
154     } catch (DOMException e) {
155         e.printStackTrace();
156     }
157 }
158

```

```

DOMDataReader.java DOMDataModifier.java hospital.xml hospital.xsd
118 } catch (DOMException e) {
119     e.printStackTrace();
120 }
121 }
122
123 public static void readPatients(Element rootElement) {
124     System.out.println("\nPatients");
125     try {
126         NodeList nodeList = rootElement.getElementsByTagName("patient");
127
128         for (int i = 0; i < nodeList.getLength(); i++) {
129             Node currentNode = nodeList.item(i);
130             if (currentNode.getNodeType() == Node.ELEMENT_NODE) {
131                 Element currentElement = (Element) currentNode;
132
133                 String id = currentElement.getAttribute("id");
134                 System.out.println("id: " + id);
135
136                 //
137                 //
138                 //
139                 //
140                 //
141                 String fullname = currentElement.getElementsByTagName("fullname").item(0).getTextContent();
142                 System.out.println("\tName: " + fullname);
143
144                 String birthdate = currentElement.getElementsByTagName("birthdate").item(0).getTextContent();
145                 System.out.println("\tBirthdate: " + birthdate);
146
147                 // multivalued
148                 for (int j = 0; j < currentElement.getElementsByTagName("allergy").getLength(); j++) {
149                     String allergy = currentElement.getElementsByTagName("allergy").item(j).getTextContent();
150                     System.out.println("\tAllergy: " + allergy);
151                 }
152             }
153         }
154     } catch (DOMException e) {
155         e.printStackTrace();
156     }
157 }
158

```



```

161 public static void readMedicines(Element rootElement) {
162     System.out.println("\nMedicines: ");
163     try {
164         NodeList nodeList = rootElement.getElementsByTagName("medicine");
165
166         for (int i = 0; i < nodeList.getLength(); i++) {
167             Node currentNode = nodeList.item(i);
168             if (currentNode.getNodeType() == Node.ELEMENT_NODE) {
169                 Element currentElement = (Element) currentNode;
170
171                 String id = currentElement.getAttribute("id");
172                 System.out.println("id: " + id);
173
174                 String name = currentElement.getElementsByTagName("name").item(0).getTextContent();
175                 System.out.println("\tName: " + name);
176
177                 String price = currentElement.getElementsByTagName("price").item(0).getTextContent();
178                 System.out.println("\tPrice: " + price);
179             }
180         }
181     } catch (DOMException e) {
182         e.printStackTrace();
183     }
184 }
185
186
187 public static void readTreatments(Element rootElement) {
188     System.out.println("\nTreatments: ");
189     try {
190         NodeList nodeList = rootElement.getElementsByTagName("treatment");
191
192         for (int i = 0; i < nodeList.getLength(); i++) {
193             Node currentNode = nodeList.item(i);
194             if (currentNode.getNodeType() == Node.ELEMENT_NODE) {
195                 Element currentElement = (Element) currentNode;
196
197                 //patient
198                 String patient = currentElement.getElementsByTagName("patient-id").item(0).getTextContent();
199                 NodeList patientNodes = rootElement.getElementsByTagName("patient");
200                 for (int j = 0; j < patientNodes.getLength(); j++) {
201                     Node currentPatientNode = patientNodes.item(j);
202                     if (currentPatientNode.getNodeType() == Node.ELEMENT_NODE) {
203                         Element currentPatientElement = (Element) currentPatientNode;

```

## 2.2 Adatmodositas - DOMDataModifier.java

```
DOMDataReader.java  DOMDataModifier.java  hospital.xml  hospital.xsd
1 package domData;
2
3 import java.io.File;
18
19 public class DOMDataModifier {
20     public static void main(String[] args) {
21
22         try {
23             File inputFile = new File("hospital.xml");
24             DocumentBuilderFactory docFactory = DocumentBuilderFactory.newInstance();
25             DocumentBuilder docBuilder = docFactory.newDocumentBuilder();
26             Document doc = docBuilder.parse(inputFile);
27
28             Element rootElement = doc.getDocumentElement();
29
30             String oldPatientID = "pat23", newPatientID = "pat11";
31             modifyPatientID(rootElement, oldPatientID, newPatientID);
32             String medicineName = "Aspirin";
33             int newPrice = 1662;
34             updateMedicinePrice(rootElement, medicineName, newPrice);
35
36             // write the content on back into the file
37             TransformerFactory transformerFactory = TransformerFactory.newInstance();
38             Transformer transformer = transformerFactory.newTransformer();
39             DOMSource source = new DOMSource(doc);
40             StreamResult outputResult = new StreamResult(inputFile);
41             transformer.transform(source, outputResult);
42
43         } catch (Exception e) {
44             e.printStackTrace();
45         }
46     }
47 }

DOMDataReader.java  DOMDataModifier.java  hospital.xml  hospital.xsd
47
48 public static void modifyPatientID(Element rootElement, String oldPatientID, String newPatientID) {
49     try {
50         NodeList patientList = rootElement.getElementsByTagName("patient");
51         // update patient id
52         for (int i = 0; i < patientList.getLength(); i++) {
53             NamedNodeMap attr = patientList.item(i).getAttributes();
54             Node nodeAttr = attr.getNamedItem("id");
55             if (oldPatientID.equalsIgnoreCase(nodeAttr.getTextContent())) {
56                 nodeAttr.setTextContent(newPatientID);
57                 System.out.println("Modified patient with id " + oldPatientID + ".");
58                 break;
59             }
60         }
61     } catch (DOMException e) {
62         e.printStackTrace();
63     }
64 }
65
66 public static void updateMedicinePrice(Element rootElement, String medicineName, int newPrice) {
67     try {
68         // all medicine nodes
69         NodeList medicineList = rootElement.getElementsByTagName("medicine");
70         // all nodes of one medicine
71         NodeList allMedicine = medicineList.item(0).getChildNodes();
72         for (int i = 0; i < allMedicine.getLength(); i++) {
73             // test if current node is the same as the one we are looking for
74             Node medNode = allMedicine.item(i);
75             if (medNode.getNodeName().equals("name") && medicineName.equalsIgnoreCase(medNode.getTextContent())) {
76                 Element medElement = (Element) medNode.getParentNode();
77                 // System.out.println(medElement.getElementsByTagName("price").item(0).getTextContent());
78                 medElement.getElementsByTagName("price").item(0).setTextContent(String.valueOf(newPrice));
79                 System.out.println("Modified price of " + medicineName + " to " + newPrice);
80                 break;
81             }
82         }
83     } catch (DOMException e) {
84         e.printStackTrace();
85     }
86 }
87 }
88 }
```