

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Készítette: Hajdu Gábor

Neptunkód: A97C75

Feladat leírása:

A beadandó feladatom témája egy gyógyszerár és annak kapcsolata a raktárakkal, a gyógyszerekkel, vásárlókkal, vezetőkkel és a vásárlókkal. Ezt fogom szemléltetni ER modellel, XDM modellel, XML és XMLSchema dokumentumokkal.

A **Gyógyszertár** mondható a központi elemnek, viszont mind ugyan olyan fontos. Megadtam a gyógyszerhárról az ID-jét, a címét és a nevét.

Mindemellett ott van még a **Raktár** a **Gyógyszertárvezető** és a **Dolgozó**, amelyek szintén ID-vel, címmel és névvel rendelkeznek. A cím tulajdonság minden esetben irányítószámot, várost, utcát, és házszámot tartalmaz.

A gyógyszerári nyilvántartásból a **Gyógyszer** sem maradhat ki. Ez rendelkezik hatóanyag, típus, név attribútummal, valamint egy tulajdonsággal ami megadja, hogy az adott gyógyszer receptre kapható-e.

Végül van **Vásárló** is, amelynek a nevét, háziorvosát és gyógyszerérzékenységeit tároljuk el.

A modell 5 kapcsolattal rendelkezik:

Szállítás, ami a raktárakat köti össze a gyógyszerhárrakkal

Vásárlás, ami a vásárlókat köti össze a gyógyszerhárrakkal

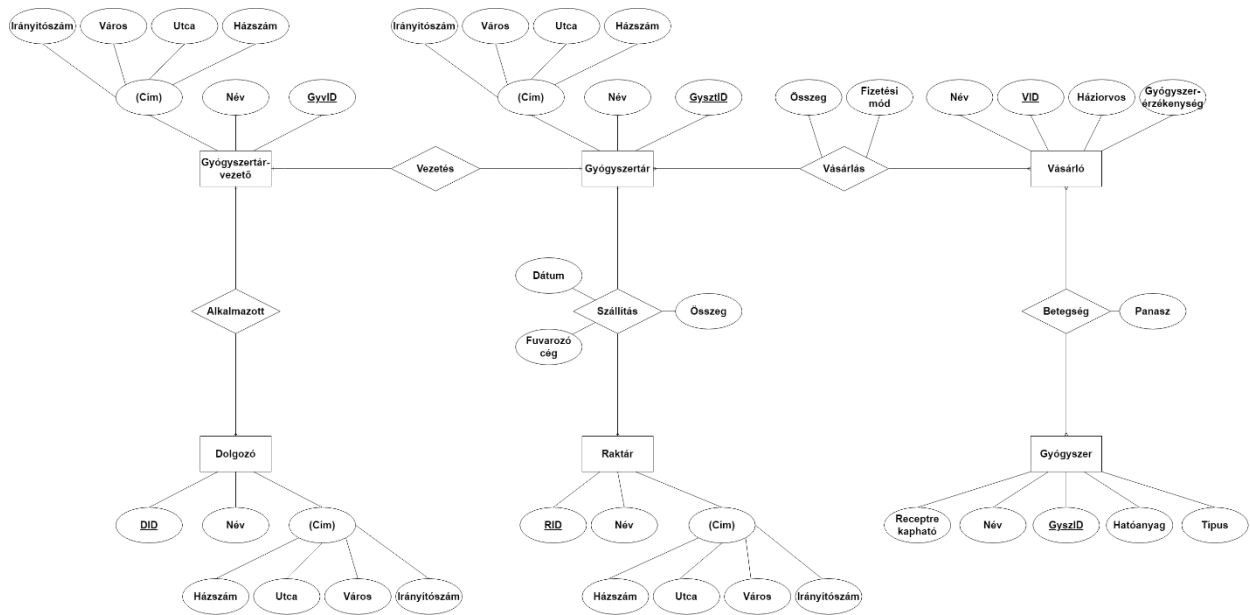
Betegség, ami a vásárlókat köti össze a gyógyszerekkel

Vezetés, ami a gyógyszerár vezetőket köti össze a gyógyszerhárrakkal

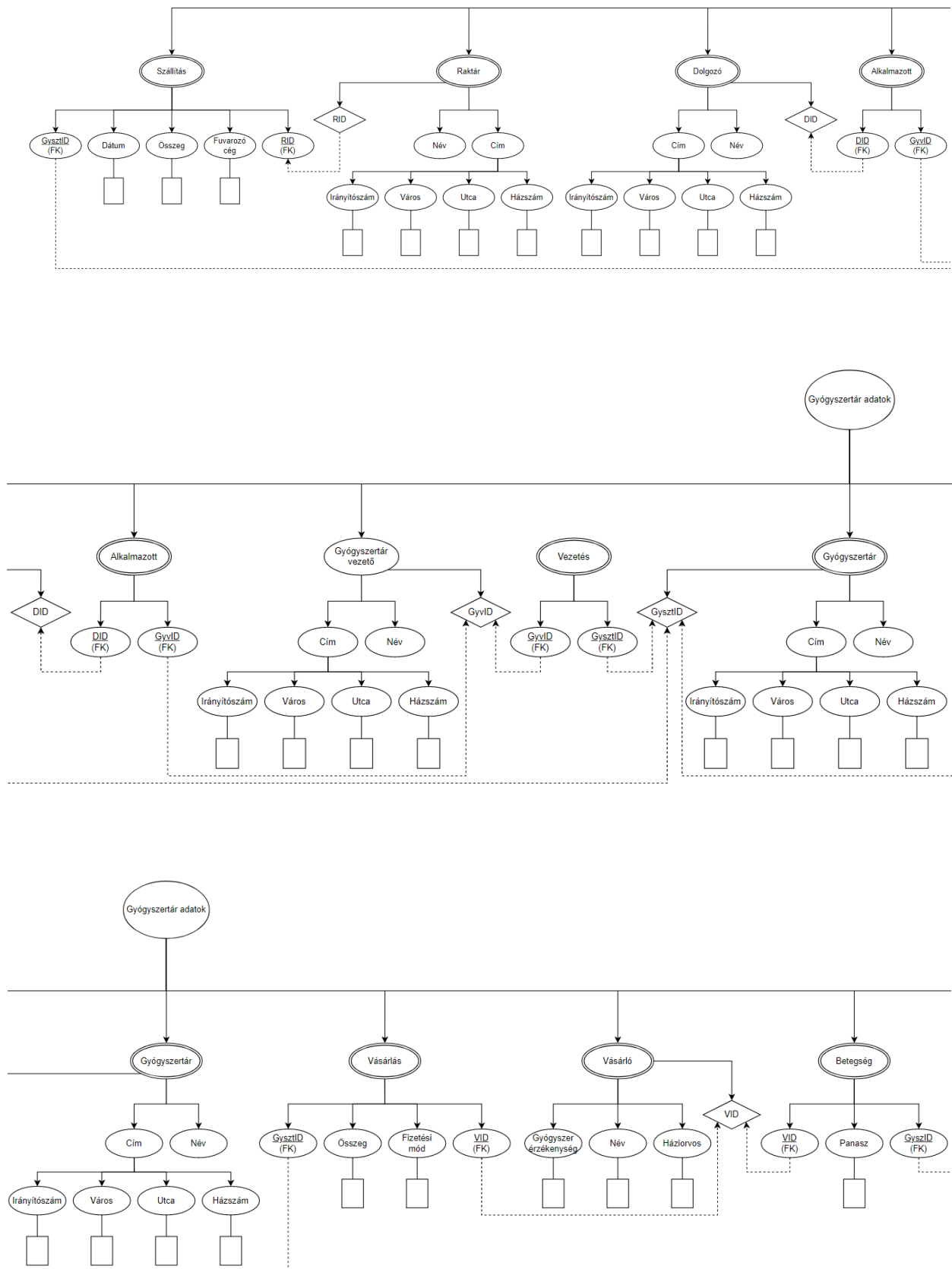
Alkalmazott, ami a gyógyszerár vezetőket köti össze a dolgozókkal

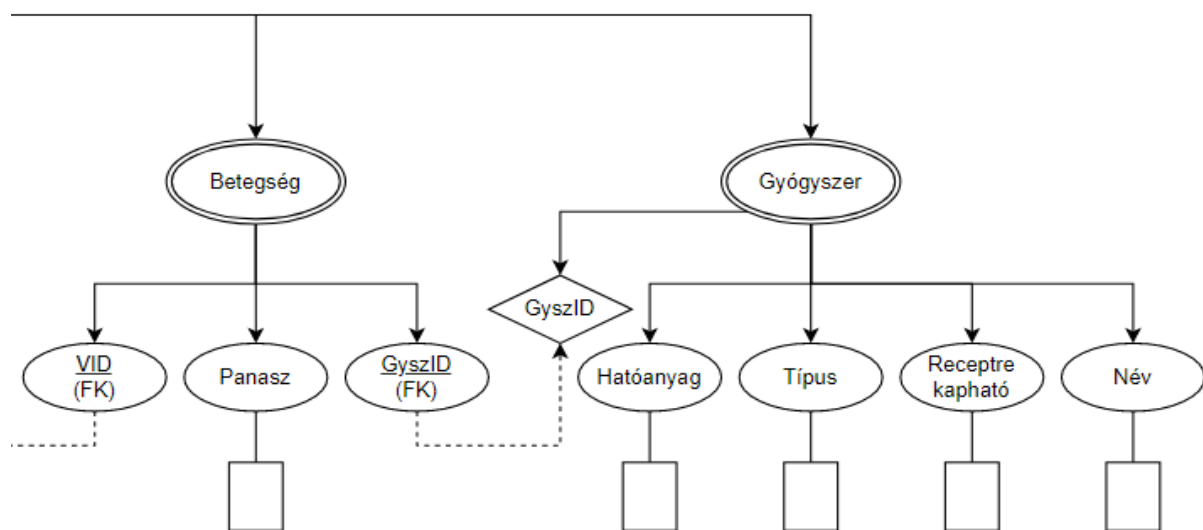
1.feladat:

1a, ER modell



1b, XDM modell





1c, XML dokumentum

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><gyogyszer_adatok xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="XMLSchemaA97C75.xsd">
```

```
<raktarak>
  <raktar RID="1">
    <cim>
      <iranyitoszam>3526</iranyitoszam>
      <varos>Miskolc</varos>
      <utca>Szentpéteri kapu</utca>
      <hazszam>47</hazszam>
    </cim>
    <nev>The Storage</nev>
  </raktar>
  <raktar RID="3">
    <cim>
      <iranyitoszam>3529</iranyitoszam>
      <varos>Miskolc</varos>
      <utca>Szentgyörgy út</utca>
      <hazszam>10</hazszam>
    </cim>
    <nev>MedStor</nev>
  </raktar>
</raktarak>

<gyogyszerarak>
  <gyogyszerar GysztID="12">
    <cim>
      <iranyitoszam>3526</iranyitoszam>
      <varos>Miskolc</varos>
      <utca>Szentpéteri kapu</utca>
      <hazszam>72-76</hazszam>
    </cim>
    <nev>Szent Péter gyógyszerár</nev>
  </gyogyszerar>
  <gyogyszerar GysztID="4">
    <cim>
      <iranyitoszam>3529</iranyitoszam>
      <varos>Miskolc</varos>
      <utca>Szentgyörgy út</utca>
      <hazszam>25</hazszam>
    </cim>
    <nev>Benu gyógyszerár</nev>
  </gyogyszerar>
</gyogyszerarak>

<gyogyszerarvezetok>
  <gyogyszerarvezeto GyvID="KP13">
    <cim>
      <iranyitoszam>3526</iranyitoszam>
      <varos>Miskolc</varos>
      <utca>Katowice</utca>
      <hazszam>14</hazszam>
    </cim>
    <nev>Dr. Kalapos Péter</nev>
  </gyogyszerarvezeto>
  <gyogyszerarvezeto GyvID="FZ2">
    <cim>
      <iranyitoszam>3529</iranyitoszam>
      <varos>Miskolc</varos>
      <utca>Áfonyás utca</utca>
      <hazszam>25</hazszam>
    </cim>
    <nev>Dr. Fekete Zoltán</nev>
  </gyogyszerarvezeto>
</gyogyszerarvezetok>

<dolgozok>
  <dolgozo DID="MF12"><cim>
    <iranyitoszam>3526</iranyitoszam>
    <varos>Miskolc</varos>
    <utca>Gömöri tér</utca>
    <hazszam>8</hazszam>
  </cim>
  <nev>Molnár Ferenc</nev>
</dolgozo>
  <dolgozo DID="SM2">
    <cim>
      <iranyitoszam>3525</iranyitoszam>
      <varos>Miskolc</varos>
      <utca>Galagonyás sor</utca>
      <hazszam>1</hazszam>
    </cim>
```

```
</cim>
<nev>Soós Márton</nev>
</dolgozo>
</dolgozok>

<gyogyszerek>
  <gyogyszer GyszID="6318">
    <receptre_kaphato>Nem</receptre_kaphato>
    <hatoanyag>metamizol-nátrium</hatoanyag>
    <tipus>fájdalom- és lázcsillapító, gyulladáscsökkentő</tipus>
    <nev>Algopyrin</nev>
  </gyogyszer>
  <gyogyszer GyszID="8113">
    <receptre_kaphato>Nem</receptre_kaphato>
    <hatoanyag>magnézium</hatoanyag>
    <tipus>Vitamin</tipus>
    <nev>Magnerot</nev>
  </gyogyszer>
</gyogyszerek>

<vasarlok>
  <vasarlo VID="983323">
    <nev>Hajdu Gábor</nev>
    <haziorvos>Dr. Lőránt Tibor</haziorvos>
    <gyogyszererzekenyseg>nincs</gyogyszererzekenyseg>
  </vasarlo>
  <vasarlo VID="64531">
    <nev>Horvát Róbert</nev>
    <haziorvos>Dr. Kovács Péter</haziorvos>
    <gyogyszererzekenyseg>metamizol-nátrium</gyogyszererzekenyseg>
  </vasarlo>
</vasarlok>

<gyogyszertar-raktar>
  <szallitas GysztID="12" RID="1">
    <datum>2020-12-03</datum>
    <fuvarozo_ceg>DHL</fuvarozo_ceg>
    <osszeg>456000</osszeg>
  </szallitas>
  <szallitas GysztID="4" RID="10">
    <datum>2020-11-10</datum>
    <fuvarozo_ceg>GLS</fuvarozo_ceg>
    <osszeg>123000</osszeg>
  </szallitas>
</gyogyszertar-raktar>

<gyogyszertar-vasarlo>
  <vasarlas GysztID="12" VID="983323">
    <osszeg>5300</osszeg>
    <fizetesi_mod>Bankkártya</fizetesi_mod>
  </vasarlas>
  <vasarlas GysztID="4" VID="64531">
    <osszeg>12340</osszeg>
    <fizetesi_mod>Készpénz</fizetesi_mod>
  </vasarlas>
</gyogyszertar-vasarlo>

<vasarlo-gyogyszer>
  <betegseg GyszID="8113" VID="64531">
    <panasz>nincs</panasz>
  </betegseg>
  <betegseg GyszID="6318" VID="983323">
    <panasz>fejfájás</panasz>
  </betegseg>
</vasarlo-gyogyszer>

<gyogyszertar-gyogyszertarvezeto>
  <vezetes GysztID="12" GyvID="KP13"/>
  <vezetes GysztID="4" GyvID="FZ2"/>
</gyogyszertar-gyogyszertarvezeto>

<dolgozo-gyogyszertarvezeto>
  <alkalmazott DID="MF12" GyvID="KP13"/>
  <alkalmazott DID="SM2" GyvID="FZ2"/>
</dolgozo-gyogyszertarvezeto>

</gyogyszertar_adatok>
```

1d, XMLSchema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema attributeFormDefault="qualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="gyogyszertar_adatok">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="raktarak">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="raktar" type="raktarTipus" maxOccurs="unbounded" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="gyogyszertarak">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="gyogyszertar" type="gyogyszertarTipus" maxOccurs="unbounded" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="gyogyszertarvezetok">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="gyogyszertarvezeto" type="gyogyszertarvezetoTipus" maxOccurs="unbounded" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="dolgozok">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="dolgozo" type="dolgozoTipus" maxOccurs="unbounded" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="gyogyszerek">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="gyogyszer" type="gyogyszerTipus" maxOccurs="unbounded" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="vasarlok">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="vasarlo" type="vasarloTipus" maxOccurs="unbounded" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="gyogyszertar-raktar">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="szallitas" type="szallitasTipus" maxOccurs="unbounded" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="gyogyszertar-vasarlo">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="vasarlas" type="vasarlasTipus" maxOccurs="unbounded" minOccurs="0"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="vasarlo-gyogyszer">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="betegseg" type="betegsegTipus" maxOccurs="unbounded" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="gyogyszertar-gyogyszertarvezeto">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="vezetes" type="vezetesTipus" maxOccurs="unbounded" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="dolgozo-gyogyszertarvezeto">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="alkalmazott" type="alkalmazottTipus" maxOccurs="unbounded" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```



```

    </xs:sequence>
</xs:complexType>

<xs:key name="gyogyszertar_ID">
  <xs:selector xpath="gyogyszertar" />
  <xs:field xpath="@GyszID"/></xs:field>
</xs:key>
<xs:key name="gyogyszertar_vezeto_ID">
  <xs:selector xpath="gyogyszertarvezeto" />
  <xs:field xpath="@GyvID"/></xs:field>
</xs:key>
<xs:key name="dolgozo_ID">
  <xs:selector xpath="dolgozo" />
  <xs:field xpath="@DID"/></xs:field>
</xs:key>
<xs:key name="raktar_ID">
  <xs:selector xpath="raktar" />
  <xs:field xpath="@RID"/></xs:field>
</xs:key>
<xs:key name="vasarLo_ID">
  <xs:selector xpath="vasarLo" />
  <xs:field xpath="@VID"/></xs:field>
</xs:key>
<xs:key name="gyogyszer_ID">
  <xs:selector xpath="gyogyszer" />
  <xs:field xpath="@GyszID"/></xs:field>
</xs:key>
<xs:keyref refer="gyogyszertar_ID"
  name="vezetes-gyogyszertar">
  <xs:selector xpath="vezetes"/></xs:selector>
  <xs:field xpath="@GyszID"/></xs:field>
</xs:keyref>
<xs:keyref refer="gyogyszertar_vezeto_ID"
  name="vezetes-gyogyszertarvezeto">
  <xs:selector xpath="vezetes"/></xs:selector>
  <xs:field xpath="@GyvID"/></xs:field>
</xs:keyref>
<xs:keyref refer="raktar_ID" name="szallitas-raktar">
  <xs:selector xpath="szallitas"/></xs:selector>
  <xs:field xpath="@RID"/></xs:field>
</xs:keyref>
<xs:keyref refer="gyogyszertar_ID"
  name="szallitas-gyogyszertar">
  <xs:selector xpath="szallitas"/></xs:selector>
  <xs:field xpath="@GyszID"/></xs:field>
</xs:keyref>
<xs:keyref refer="gyogyszertar_vezeto_ID"
  name="alkalmazott-gyogyszertarvezeto">
  <xs:selector xpath="alkalmazott"/></xs:selector>
  <xs:field xpath="@GyvID"/></xs:field>
</xs:keyref>
<xs:keyref refer="dolgozo_ID" name="alkalmazott-dolgozo">
  <xs:selector xpath="alkalmazott"/></xs:selector>
  <xs:field xpath="@DID"/></xs:field>
</xs:keyref>
<xs:keyref refer="gyogyszertar_ID"
  name="vasarLas-gyogyszertar">
  <xs:selector xpath="vasarLas"/></xs:selector>
  <xs:field xpath="@GyszID"/></xs:field>
</xs:keyref>
<xs:keyref refer="vasarLo_ID" name="vasarLas-vasarLo">
  <xs:selector xpath="vasarLas"/></xs:selector>
  <xs:field xpath="@VID"/></xs:field>
</xs:keyref>
<xs:keyref refer="gyogyszer_ID" name="betegseg-gyogyszer">
  <xs:selector xpath="betegseg"/></xs:selector>
  <xs:field xpath="@GyszID"/></xs:field>
</xs:keyref>
<xs:keyref refer="vasarLo_ID" name="betegseg-vasarLo">
  <xs:selector xpath="betegseg"/></xs:selector>
  <xs:field xpath="@VID"/></xs:field>
</xs:keyref>
</xs:element>

<xs:element name="cim">
  <xs:complexType>
    <xs:sequence>
      <xs:element type="xs:short" name="iranyitoszam" />
      <xs:element type="xs:string" name="varos" />
      <xs:element type="xs:string" name="utca" />
      <xs:element type="xs:string" name="hazszam" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

    </xs:complexType>
  </xs:element>

  <xs:element name="receptre_kaphato">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="Igen" />
        <xs:enumeration value="Nem" />
      </xs:restriction>
    </xs:simpleType>
  </xs:element>

  <xs:element name="fizetesi_mod">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="Bankkártya" />
        <xs:enumeration value="Készpénz" />
        <xs:enumeration value="SZÉP-kártya" />
      </xs:restriction>
    </xs:simpleType>
  </xs:element>

  <xs:simpleType name="id">
    <xs:restriction base="xs:int">
      <xs:minInclusive value="1" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="munka_id">
    <xs:restriction base="xs:string">
      <xs:pattern value="[A-Z]+[A-Z]+[0-9]+[0-9]" />
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name='raktarTipus'>
    <xs:sequence>
      <xs:element ref="cim" />
      <xs:element type="xs:string" name="nev" />
    </xs:sequence>
    <xs:attribute type="id" name="RID" />
  </xs:complexType>

  <xs:complexType name='gyogyszerTartarTipus'>
    <xs:sequence>
      <xs:element ref="cim" />
      <xs:element type="xs:string" name="nev" />
    </xs:sequence>
    <xs:attribute type="id" name="GyszID" />
  </xs:complexType>

  <xs:complexType name='gyogyszerTartarvezetoTipus'>
    <xs:sequence>
      <xs:element ref="cim" />
      <xs:element type="xs:string" name="nev" />
    </xs:sequence>
    <xs:attribute type="munka_id" name="GyvID" />
  </xs:complexType>

  <xs:complexType name='dolgozoTipus'>
    <xs:sequence>
      <xs:element ref="cim" />
      <xs:element type="xs:string" name="nev" />
    </xs:sequence>
    <xs:attribute type="munka_id" name="DID" />
  </xs:complexType>

  <xs:complexType name="gyogyszerTipus">
    <xs:sequence>
      <xs:element ref="receptre_kaphato" />
      <xs:element type="xs:string" name="hatoanyag" />
      <xs:element type="xs:string" name="tipus" />
      <xs:element type="xs:string" name="nev" />
    </xs:sequence>
    <xs:attribute type="id" name="GyszID" />
  </xs:complexType>

  <xs:complexType name="vasarLoTipus">
    <xs:sequence>
      <xs:element type="xs:string" name="nev" />
      <xs:element type="xs:string" name="haziorvos" />
      <xs:element type="xs:string" name="gyogyszererzekenyseg" />
    </xs:sequence>
    <xs:attribute type="id" name="VID" />
  </xs:complexType>

```

```
<xs:attribute type="id" name="VID" />
</xs:complexType>

<xs:complexType name="szallitasTipus">
  <xs:sequence>
    <xs:element type="xs:date" name="datum" />
    <xs:element type="xs:string" name="fuvarozo_ceg" />
    <xs:element type="xs:int" name="osszeg" />
    <xs:element type="id" name="GysztID" />
    <xs:element type="id" name="RID" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="vasarlasTipus">
  <xs:sequence>
    <xs:element type="xs:int" name="osszeg" />
    <xs:element ref="fizetes_mod" />
    <xs:element type="id" name="GysztID" />
    <xs:element type="id" name="VID" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="betegsegTipus">
  <xs:sequence>
    <xs:element type="xs:string" name="panasz" />
    <xs:element type="id" name="VID" />
    <xs:element type="id" name="GyszID" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="vezetesTipus">
  <xs:sequence>
    <xs:element type="munka_id" name="GyvID" />
    <xs:element type="id" name="GysztID" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="alkalmazottTipus">
  <xs:sequence>
    <xs:element type="munka_id" name="GyvID" />
    <xs:element type="munka_id" name="DID" />
  </xs:sequence>
</xs:complexType>
</xs:schema>
```

2.feladat:

2a, DOM adatolvasás

```
package hu.domparse.a97c75;

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

public class DomReadA97C75 {

    public static void main(String[] args)

        throws ParserConfigurationException, IOException, SAXException {

        File xml = new File("XMLA97C75.xml"); //Az XML file beolvasása
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance(); //DocumentBuilderFactory inicializálása

        DocumentBuilder docBuilder = factory.newDocumentBuilder(); // A builder példányosítása
        Document document = docBuilder.parse(xml); //A dokumentum konvertálása hogy ártse az xml-t
        document.getDocumentElement().normalize();

        ReadFile(document); //Kijratjuk az összes adatot
    }

    public static void ReadFile(Document document) { //Az adatok beolvasása és kiírása

        NodeList nodeList1 = document.getElementsByTagName("raktar"); //Lista létrehozása a gyermekelemeknek a raktár kulcsszóra
        for (int i = 0; i < nodeList1.getLength(); i++) {
            Node rNode = nodeList1.item(i);

            if (rNode.getNodeType() == Node.ELEMENT_NODE) { //Elemek id-hez adása
                Element element1 = (Element) rNode;
                String raktar_id = element1.getAttribute("RID"); //Attribútumok eltárolása

                Node rNode1 = element1.getElementsByTagName("iranyitoszam").item(0); //Elemek eltárolása
                String irányitoszam = rNode1.getTextContent();

                Node rNode2 = element1.getElementsByTagName("varos").item(0);
                String varos = rNode2.getTextContent();

                Node rNode3 = element1.getElementsByTagName("utca").item(0);
                String utca = rNode3.getTextContent();

                Node rNode4 = element1.getElementsByTagName("hazszam").item(0);
                String hazszam = rNode4.getTextContent();

                System.out.println("Raktár adatok:\n\tRaktár ID: " + raktar_id + ", Cím: (Irányítószám: "
                    + irányitoszam + ", Város: " + varos + ", Utca: " + utca + ", Házszám: " + hazszam + ").\n"); //Az eltárolt adatok kiírása
            }
        }

        NodeList nodeList2 = document.getElementsByTagName("gyogyszertar");
        for (int i = 0; i < nodeList2.getLength(); i++) {
            Node gytNode = nodeList2.item(i);

            if (gytNode.getNodeType() == Node.ELEMENT_NODE) {
                Element element2 = (Element) gytNode;
                String gyogyszertar_id = element2.getAttribute("GysztID");

                Node gysztNode1 = element2.getElementsByTagName("iranyitoszam").item(0);
                String irányitoszam = gysztNode1.getTextContent();

                Node gysztNode2 = element2.getElementsByTagName("varos").item(0);
                String varos = gysztNode2.getTextContent();

                Node gysztNode3 = element2.getElementsByTagName("utca").item(0);
                String utca = gysztNode3.getTextContent();

                Node gysztNode4 = element2.getElementsByTagName("hazszam").item(0);
                String hazszam = gysztNode4.getTextContent();

                Node gysztNode5 = element2.getElementsByTagName("nev").item(0);
                String nev = gysztNode5.getTextContent();

                System.out.println("Gyógyszertár adatok:\n\tGyógyszertár ID: " + gyogyszertar_id + ", Gyógyszertár név: " + nev +
                    ", Cím: (Irányítószám: " + irányitoszam + ", Város: " + varos + ", Utca: " + utca + ", Házszám: " + hazszam + ").\n");
            }
        }
    }
}
```

```

NodeList nodeList3 = document.getElementsByTagName("gyogyszer");
for (int i = 0; i < nodeList3.getLength(); i++) {
    Node gyNode = nodeList3.item(i);

    if (gyNode.getNodeType() == Node.ELEMENT_NODE) {
        Element element3 = (Element) gyNode;
        String gyogyszer_id = element3.getAttribute("GyszID");

        Node gyszNode1 = element3.getElementsByTagName("receptre_kaphato").item(0);
        String receptre_kaphato = gyszNode1.getTextContent();

        Node gyszNode2 = element3.getElementsByTagName("hatoanyag").item(0);
        String hatoanyag = gyszNode2.getTextContent();

        Node gyszNode3 = element3.getElementsByTagName("tipus").item(0);
        String tipus = gyszNode3.getTextContent();

        Node gyszNode4 = element3.getElementsByTagName("nev").item(0);
        String nev = gyszNode4.getTextContent();

        System.out.println("Gyógyszer adatok:\n\tGyógyszer ID: " + gyogyszer_id + ", Név: " + nev +
            ", Receptre kapható: " + receptre_kaphato + ", Hatóanyag: " + hatoanyag + ", Típus: " + tipus + ".\n");
    }
}

NodeList nodeList4 = document.getElementsByTagName("vasarlo");
for (int i = 0; i < nodeList4.getLength(); i++) {
    Node vNode = nodeList4.item(i);

    if (vNode.getNodeType() == Node.ELEMENT_NODE) {
        Element element4 = (Element) vNode;
        String vasarlo_id = element4.getAttribute("VID");

        Node vNode1 = element4.getElementsByTagName("nev").item(0);
        String nev = vNode1.getTextContent();

        Node vNode2 = element4.getElementsByTagName("haziorvos").item(0);
        String haziorvos = vNode2.getTextContent();

        Node vNode3 = element4.getElementsByTagName("gyogyszererzekenyseg").item(0);
        String gyogyszererzekenyseg = vNode3.getTextContent();

        System.out.println("Vásárló adatok:\n\tVásárló ID: " + vasarlo_id + ", Név: " + nev + ", Házi orvos: " + haziorvos +
            ", Gyógyszerérzékenység: " + gyogyszererzekenyseg + ".\n");
    }
}

NodeList nodeList5 = document.getElementsByTagName("dolgozo");
for (int i = 0; i < nodeList5.getLength(); i++) {
    Node rNode = nodeList5.item(i);

    if (rNode.getNodeType() == Node.ELEMENT_NODE) {
        Element element5 = (Element) rNode;
        String dolgozo_id = element5.getAttribute("DID");

        Node dNode1 = element5.getElementsByTagName("nev").item(0);
        String nev = dNode1.getTextContent();

        Node dNode2 = element5.getElementsByTagName("iranyitoszam").item(0);
        String iranyitoszam = dNode2.getTextContent();

        Node dNode3 = element5.getElementsByTagName("varos").item(0);
        String varos = dNode3.getTextContent();

        Node dNode4 = element5.getElementsByTagName("utca").item(0);
        String utca = dNode4.getTextContent();

        Node dNode5 = element5.getElementsByTagName("hazszam").item(0);
        String hazszam = dNode5.getTextContent();

        System.out.println("Dolgozó adatok:\n\tDolgozó ID: " + dolgozo_id + ", Név: " + nev +
            ", Cím: (Iránítószám: " + iranyitoszam + ", Város: " + varos + ", Utca: " + utca + ", Házzám: " + hazszam + ").\n");
    }
}

NodeList nodeList6 = document.getElementsByTagName("gyogyszertarvezeto");
for (int i = 0; i < nodeList6.getLength(); i++) {
    Node rNode = nodeList6.item(i);

    if (rNode.getNodeType() == Node.ELEMENT_NODE) {
        Element element6 = (Element) rNode;
        String gyogyszertar_vezeto_id = element6.getAttribute("GyvID");

        Node gyvNode1 = element6.getElementsByTagName("nev").item(0);
    }
}

```

```

        String nev = gyvNode1.getTextContent();

        Node gyvNode2 = element6.getElementsByTagName("iranyitoszam").item(0);
        String iranyitoszam = gyvNode2.getTextContent();

        Node gyvNode3 = element6.getElementsByTagName("varos").item(0);
        String varos = gyvNode3.getTextContent();

        Node gyvNode4 = element6.getElementsByTagName("utca").item(0);
        String utca = gyvNode4.getTextContent();

        Node gyvNode5 = element6.getElementsByTagName("hazszam").item(0);
        String hazszam = gyvNode5.getTextContent();

        System.out.println("Gyógyszertár vezető adatok:\n\tGyógyszertár vezető ID: " + gyogyszertar_vezeto_id + ", Név: " + nev +
        ", Cím: (Irányítószám: " + iranyitoszam + ", Város: " + varos + ", Utca: " + utca + ", Házzám: " + hazszam + ").\n");
    }
}

NodeList nodeList7 = document.getElementsByTagName("szallitas");
for (int i = 0; i < nodeList7.getLength(); i++) {
    Node szNode = nodeList7.item(i);

    if (szNode.getNodeType() == Node.ELEMENT_NODE) {
        Element element7 = (Element) szNode;
        String gyogyszertar_id = element7.getAttribute("GysztID");
        String raktar_id = element7.getAttribute("RID");

        Node szNode1 = element7.getElementsByTagName("datum").item(0);
        String datum = szNode1.getTextContent();

        Node szNode2 = element7.getElementsByTagName("fuvarozo_ceg").item(0);
        String fuvarozo_ceg = szNode2.getTextContent();

        Node szNode3 = element7.getElementsByTagName("osszeg").item(0);
        String osszeg = szNode3.getTextContent();

        System.out.println("Szállítási adatok:\n\tGyógyszertár ID: " + gyogyszertar_id + ", Raktár ID: " + raktar_id + ", Dátum: " + datum
        + ", Fuvarozó cég: " + fuvarozo_ceg + ", Összeg: " + osszeg + ".\n");
    }
}

NodeList nodeList8 = document.getElementsByTagName("vasarlas");
for (int i = 0; i < nodeList8.getLength(); i++) {
    Node vNode = nodeList8.item(i);

    if (vNode.getNodeType() == Node.ELEMENT_NODE) {
        Element element8 = (Element) vNode;
        String gyogyszertar_id = element8.getAttribute("GysztID");
        String vasarlo_id = element8.getAttribute("VID");

        Node vNode1 = element8.getElementsByTagName("osszeg").item(0);
        String osszeg = vNode1.getTextContent();

        Node vNode2 = element8.getElementsByTagName("fizetesi_mod").item(0);
        String fizetesi_mod = vNode2.getTextContent();

        System.out.println("Vásárlási adatok:\n\tGyógyszertár ID: " + gyogyszertar_id + ", Vásárló ID: " + vasarlo_id + ", Összeg: " + osszeg
        + ", Fizetési mód: " + fizetesi_mod + ".\n");
    }
}

NodeList nodeList9 = document.getElementsByTagName("betegseg");
for (int i = 0; i < nodeList9.getLength(); i++) {
    Node bNode = nodeList9.item(i);

    if (bNode.getNodeType() == Node.ELEMENT_NODE) {
        Element element9 = (Element) bNode;
        String vasarlo_id = element9.getAttribute("VID");
        String gyogyszer_id = element9.getAttribute("GyszID");

        Node bNode1 = element9.getElementsByTagName("panasz").item(0);
        String panasz = bNode1.getTextContent();

        System.out.println("Betegség adatok:\n\tVásárló ID: " + vasarlo_id + ", Gyógyszer ID: " + gyogyszer_id + ", Panasz: " + panasz + ".\n");
    }
}

NodeList nodeList10 = document.getElementsByTagName("vezetes");
for (int i = 0; i < nodeList10.getLength(); i++) {
    Node vNode = nodeList10.item(i);

    if (vNode.getNodeType() == Node.ELEMENT_NODE) {
        Element element10 = (Element) vNode;
        String gyogyszertar_id = element10.getAttribute("GysztID");
        String gyogyszertar_vezeto_id = element10.getAttribute("GyvID");

        System.out.println("Vezetés adatok:\n\tGyógyszertár ID: " + gyogyszertar_id + ", Gyógyszertár vezető ID: " + gyogyszertar_vezeto_id + ".\n");
    }
}

NodeList nodeList11 = document.getElementsByTagName("alkalmazott");
for (int i = 0; i < nodeList11.getLength(); i++) {
    Node vNode = nodeList11.item(i);

    if (vNode.getNodeType() == Node.ELEMENT_NODE) {
        Element element11 = (Element) vNode;
        String dolgozo_id = element11.getAttribute("DID");
        String gyogyszertar_vezeto_id = element11.getAttribute("GyvID");

        System.out.println("Alkalmazott adatok:\n\tGyógyszertár vezető ID: " + gyogyszertar_vezeto_id + ", Dolgozó ID: " + dolgozo_id + ".\n");
    }
}
}
}

```

2b,DOM módosítás

```
package hu.domparse.a97c75;

import java.io.*;

public class DomModifyA97C75 {

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

        String xmlFilePath = "XMLA97C75.xml"; //File beolvasása
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder builder = factory.newDocumentBuilder();
        Document document = builder.parse(new File(xmlFilePath));

        Node masodikGyogyszer = document.getElementsByTagName("gyogyszer").item(1); //A 2. gyogyszer elem gyerekelemeinek megkeresése
        NodeList masodikGyogyszerNodeList = masodikGyogyszer.getChildNodes();

        for (int i = 0; i < masodikGyogyszerNodeList.getLength(); i++) { //A 2. gyogyszer elem hatóanyagának megváltoztatása acetilszalicilsavra
            Node element = masodikGyogyszerNodeList.item(i);
            if ("hatóanyag".equals(element.getNodeName())) {
                element.setTextContent("acetilszalicilsav");
            }
        }

        Element ar = document.createElement("Ar"); //A 2. gyogyszer elemhez hozzáadunk egy Ar elemet is.
        ar.appendChild(document.createTextNode("1759"));
        masodikGyogyszer.appendChild(ar);

        Node elsoDolgozo = document.getElementsByTagName("dolgozo").item(0); //Az 1. dolgozó elem gyerekelemeinek megkeresése
        NodeList elsoDolgozoNodeList = elsoDolgozo.getChildNodes();

        for (int i = 0; i < elsoDolgozoNodeList.getLength(); i++) { //A 1. dolgozó elem hátszámának megváltoztatása 42-re
            Node element = elsoDolgozoNodeList.item(i);
            if ("cim".equals(element.getNodeName())) {
                NodeList elementNodeList = element.getChildNodes();
                for (int j = 0; j < elementNodeList.getLength(); j++){
                    Node element2 = elementNodeList.item(j);
                    if ("hátszám".equals(element2.getNodeName())) {
                        element2.setTextContent("42");
                    }
                }
            }
        }

        TransformerFactory transformerFactory = TransformerFactory.newInstance(); //A módosításokat feltöltjük az XML file-ba
        Transformer transformer = transformerFactory.newTransformer();
        DOMSource source = new DOMSource(document);
        StreamResult result = new StreamResult(new File(xmlFilePath));
        transformer.transform(source, result);

        System.out.println("Változtatások sikeresen eszközölve az XML file-on!!!!"); //Nyugtázzuk az elvégzett módosításokat
    }
}
```

2c, DOM Lekérdezés

```
package hu.domparse.a97c75;

import java.io.File;

public class DomQueryA97C75 {

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

        File file = new File("XMLA97C75.xml"); //Az XML file beolvasása

        DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance(); //DocumentBuilderFactory inicializálása
        DocumentBuilder dBuilder = dbFactory.newDocumentBuilder(); //A builder példányosítása

        Document doc = dBuilder.parse(file); //A dokumentum konvertálása hogy értsé az xml-t
        doc.getDocumentElement().normalize();
        System.out.print("Gyökér elem: "); //A gyökérelem kiírása
        System.out.println(doc.getDocumentElement().getNodeName());

        NodeList nList = doc.getElementsByTagName("raktar"); //A raktárak mentése listába

        for (int i = 0; i < nList.getLength(); i++) { //A raktárak gyerek elemeinek megjelenítése a konzolon
            Node node = nList.item(i);
            System.out.println("\nAktuális elem : "+node.getNodeName());
            if (node.getNodeType() == Node.ELEMENT_NODE) {
                Element elem = (Element) node;
                System.out.println("ID: "+elem.getAttribute("RID"));
                NodeList nList2 = elem.getChildNodes();
                for (int j = 0; j < nList2.getLength(); j++) {
                    Node node2 = nList2.item(j);
                    if (node2.getNodeType() == Node.ELEMENT_NODE && node2.getChildNodes().getLength() == 1) {
                        System.out.println(node2.getNodeName()+" : "+node2.getTextContent());
                    }
                    else {
                        if (node2.getNodeType() == Node.ELEMENT_NODE) {
                            System.out.println(node2.getNodeName()+".");
                        }
                        NodeList nList3 = node2.getChildNodes();
                        for (int k = 0; k < nList3.getLength(); k++) {
                            Node node3 = nList3.item(k);
                            if (node3.getNodeType() == Node.ELEMENT_NODE) {
                                System.out.println("\t"+node3.getNodeName()+" : "+node3.getTextContent());
                            }
                        }
                    }
                }
            }
        }

        nList = doc.getElementsByTagName("gyogyszertar");

        for (int i = 0; i < nList.getLength(); i++) { //A gyógyszerterek gyerek elemeinek megjelenítése a konzolon
            Node node = nList.item(i);
            System.out.println("\nAktuális elem : "+node.getNodeName());
            if (node.getNodeType() == Node.ELEMENT_NODE) {
                Element elem = (Element) node;
                System.out.println("ID: "+elem.getAttribute("GysztID"));
                NodeList nList2 = elem.getChildNodes();
                for (int j = 0; j < nList2.getLength(); j++) {
                    Node node2 = nList2.item(j);
                    if (node2.getNodeType() == Node.ELEMENT_NODE && node2.getChildNodes().getLength() == 1) {
                        System.out.println(node2.getNodeName()+" : "+node2.getTextContent());
                    }
                    else {
                        if (node2.getNodeType() == Node.ELEMENT_NODE) {
                            System.out.println(node2.getNodeName()+".");
                        }
                        NodeList nList3 = node2.getChildNodes();
                        for (int k = 0; k < nList3.getLength(); k++) {
                            Node node3 = nList3.item(k);
                            if (node3.getNodeType() == Node.ELEMENT_NODE) {
                                System.out.println("\t"+node3.getNodeName()+" : "+node3.getTextContent());
                            }
                        }
                    }
                }
            }
        }

        nList = doc.getElementsByTagName("gyogyszer");

        for (int i = 0; i < nList.getLength(); i++) { //A gyógyszerek gyerek elemeinek megjelenítése a konzolon
            Node node = nList.item(i);
            System.out.println("\nAktuális elem : "+node.getNodeName());
            if (node.getNodeType() == Node.ELEMENT_NODE) {
                Element elem = (Element) node;
```



```

System.out.println("ID:"+elem.getAttribute("GyszID"));
NodeList nList2 = elem.getChildNodes();
Node node = nList.item(i);
for (int j = 0; j < nList2.getLength(); j++) {
    Node node2 = nList2.item(j);
    if (node2.getNodeType() == Node.ELEMENT_NODE && node2.getChildNodes().getLength() == 1) {
        System.out.println(node2.getNodeName()+" : "+node2.getTextContent());
    }
    else {
        if (node2.getNodeType() == Node.ELEMENT_NODE) {
            System.out.println(node2.getNodeName()+":");
        }
        NodeList nList3 = node2.getChildNodes();
        for (int k = 0; k < nList3.getLength(); k++) {
            Node node3 = nList3.item(k);
            if (node3.getNodeType() == Node.ELEMENT_NODE) {
                System.out.println("\t"+node3.getNodeName()+" : "+node3.getTextContent());
            }
        }
    }
}
}
nList = doc.getElementsByTagName("gyogyszertervezetok");

for (int i = 0; i < nList.getLength(); i++) { //A gyógyszertervezetek nyelvek elemeinek megjelenítése a konzolon
    Node node = nList.item(i);
    System.out.println("\nAktuális elem : "+node.getNodeName());
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element elem = (Element) node;
        System.out.println("ID:"+elem.getAttribute("GyvID"));
        NodeList nList2 = elem.getChildNodes();
        for (int j = 0; j < nList2.getLength(); j++) {
            Node node2 = nList2.item(j);
            if (node2.getNodeType() == Node.ELEMENT_NODE && node2.getChildNodes().getLength() == 1) {
                System.out.println(node2.getNodeName()+" : "+node2.getTextContent());
            }
            else {
                if (node2.getNodeType() == Node.ELEMENT_NODE) {
                    System.out.println(node2.getNodeName()+":");
                }
                NodeList nList3 = node2.getChildNodes();
                for (int k = 0; k < nList3.getLength(); k++) {
                    Node node3 = nList3.item(k);
                    if (node3.getNodeType() == Node.ELEMENT_NODE) {
                        System.out.println("\t"+node3.getNodeName()+" : "+node3.getTextContent());
                    }
                }
            }
        }
    }
}
nList = doc.getElementsByTagName("dolgozo");

for (int i = 0; i < nList.getLength(); i++) { //A dolgozók nyelvek elemeinek megjelenítése a konzolon
    Node node = nList.item(i);
    System.out.println("\nAktuális elem : "+node.getNodeName());
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element elem = (Element) node;
        System.out.println("ID:"+elem.getAttribute("DID"));
        NodeList nList2 = elem.getChildNodes();
        for (int j = 0; j < nList2.getLength(); j++) {
            Node node2 = nList2.item(j);
            if (node2.getNodeType() == Node.ELEMENT_NODE && node2.getChildNodes().getLength() == 1) {
                System.out.println(node2.getNodeName()+" : "+node2.getTextContent());
            }
            else {
                if (node2.getNodeType() == Node.ELEMENT_NODE) {
                    System.out.println(node2.getNodeName()+":");
                }
                NodeList nList3 = node2.getChildNodes();
                for (int k = 0; k < nList3.getLength(); k++) {
                    Node node3 = nList3.item(k);
                    if (node3.getNodeType() == Node.ELEMENT_NODE) {
                        System.out.println("\t"+node3.getNodeName()+" : "+node3.getTextContent());
                    }
                }
            }
        }
    }
}

```