

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Levelező Féléves Feladat

Korház beteg nyilvántartó rendszere

Készítette: Kiss Máté

Neptunkód: NXYCH1

Tartalomjegyzék

1. feladat	2
1.a. ER model leírása:	2
1.a.1. ER Model:	3
1.b. ER model átkonvertálása XDM modellre:	3
1.b.1. XDM modell:	4
1.c. XML dokumentum leírás:	4
1.c.1. XML dokumentum:	4
1.d. XSD-schema leírás:	6
1.d.1. XSD-schema:	7
2. Feladat	9
2.1. DOMREAD leírása:	9
2.1.1. DOMRead:	9
2.2. DOMQuery leírása:	12
2.2.1. DOMQuery:	13
2.3. DOMModify leírása:	15
2.3.1. DOMModify:	16

A feladat leírása:

A féléves feladatom első része egy kórházi beteg nyilvántartó rendszert ábrázol. Ehez tartozik egy ER model, XDM model konverzió az ER modellről, egy XML dokumentum, amiben az adatok vannak ábrázolva,

XMLSchema, ami az XML dokumentum schematikkus ábrázolása.

A feladatom második részében Java Eclipse-ben DOM használatával az elkészített XML dokumentum beolvasása, bizonyos leszűrt adatok lekérdezése, illetve konkrét adat módosítása történik.

1. feladat

1.a. ER model leírása:

Az ER model egy beteg nyilvántartó rendszert ábrázol, aminek elemei a Gyógyszerek, Kórház, Kortörténet, Beteg és az Doktor

A Gyógyszerek és Kórház között N:M kapcsolat áll fent mivel egy gyógyszer fajta több kórházban is lehet, illetve egy kórházban több fajta gyógyszer is van.

A Kórház és Beteg közt N:M kapcsolat van mivel egy kórházban lehet több beteg is és egy beteg lehet több kórházban is a valóságban attól függően, hogy az adott létesítmény alkalmas-e vagy alkalmatlan az adott panaszának ellátására.

A Kórházak és Doktorok között szintén N:M kapcsolat van mivel egy Doktor dolgozhat több Kórházban is és egy Kórházban több doktor is dolgozik.

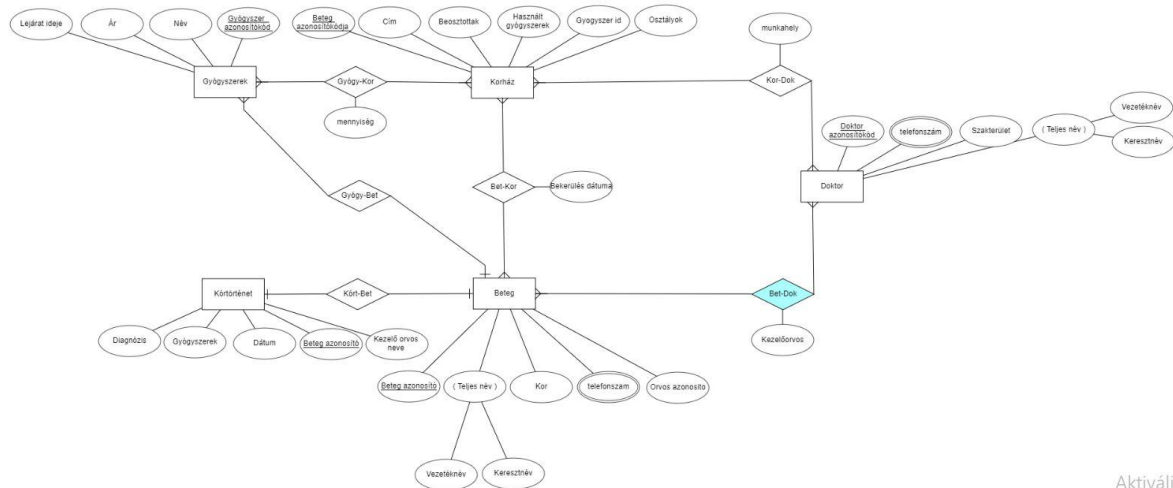
A Beteg és Doktor között szintén N:M kapcsolat van mivel egy Beteget több orvos is kezelhet és egy Orvos több beteget kezel.

A Beteg és Kortörténet között 1:1 kapcsolat található mivel egy beteghez egy kortörténet kapcsolható és fordítva is ez igaz mivel egy kortörténet csak egy betegé lehet.

A Beteg és Gyógyszerek között egy-több kapcsolat van mivel egy beteg több gyógyszert is szedhet.

Minden N:M kapcsolat rendelkezik tulajdonsággal, amit az elsődleges és idegen kulccsal azonosíthatunk be.

1.a.1. ER Model:



Aktiválja a W
A társaság a W

1.b. ER model átkonvertálása XDM modellre:

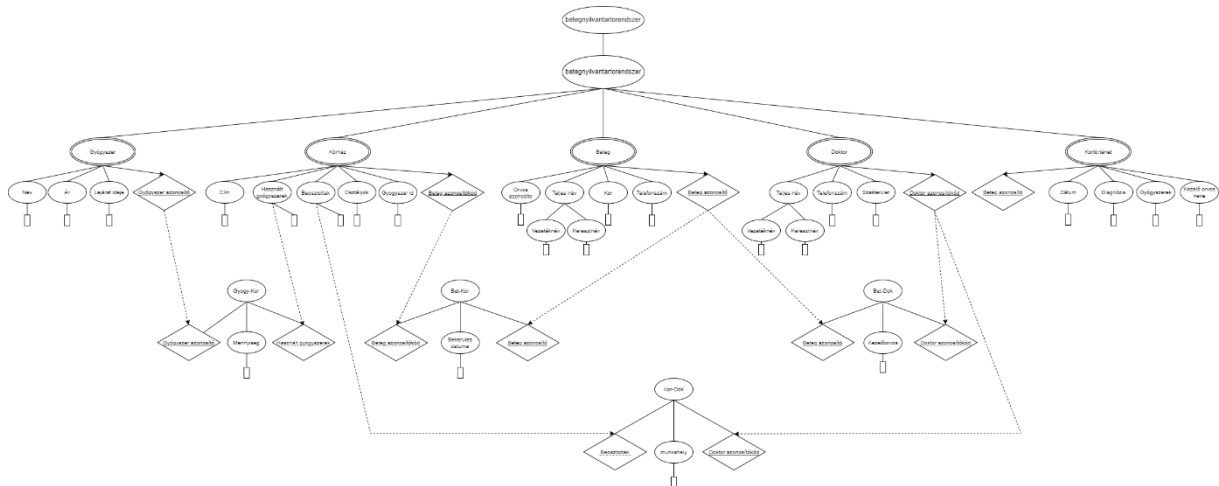
ER modellt XDM modellre úgy konvertálhatunk, hogy új jelöléseket kell bevezetni. Ezek az elem, attribútum és a szöveg. Elem az XDM modellben ellipszissel van rajzolva. Az attribútum, jelölése rombusz. Szöveges tartalmat téglalappal jelöljük.

Ezekon kívül, ha egy elem többszörös előfordulási lehetőséggel bír azt dupla ellipszissel jelöljük. A több a többhöz kapcsolatokat külön ábrázolni kell!

Figyelve a pontos elrendezésre az átláthatóság érdekében.

Jelen esetben az öt fő elem csoport egy szintbe kerül és azon elemek tulajdonságai egy szinttel lejjebb szintén egy síkba kerültek. A N:M kapcsolatok az átláthatóság érdekében lettek elrendezve, a szint eltérés a vonal kereszteződés elkerülése érdekében történt.

1.b.1. XDM modell:



1.c. XML dokumentum leírás:

Az XML dokumentumban történtek a korábban ábrázolt egyedek és tulajdonságok feltöltései. A többszörösen előforduló egyedekből legalább 3 különböző tartalmú került létrehozásra. A különböző egyedek és azok tulajdonságai között komment jelöli a váltást.

1.c.1. XML dokumentum:

```
XMLNXYCH1.xml x XMLSchemaNXYCH1.xsd DOMReadNXYCH1.java DOMQUERYNXYCH1.java DomModifyNXYCH1.java
1 <?xml version="1.0" encoding="UTF-8"?>
2 <betegnyilvantartorendszer>
3   <betegnyilvantartas>
4     <!--beteg 1-->
5     <beteg id="1">
6       <teljes_nev>
7         <vezeteknev>Kiss</vezeteknev>
8         <keresztnev>Mate</keresztnev>
9       </teljes_nev>
10      <kor>23</kor>
11      <orvos_azonosito>1</orvos_azonosito>
12      <telefonszam>06307032146</telefonszam>
13      <beteg_azonositokod>1</beteg_azonositokod>
14      <bekerules_datuma>2022-11-19</bekerules_datuma>
15      <kezeloorvos>dr.Kitalalt Ember</kezeloorvos>
16    </beteg>
17    <!--beteg 2-->
18    <beteg id="2">
19      <teljes_nev>
20        <vezeteknev>Kiss</vezeteknev>
21        <keresztnev>Antal</keresztnev>
22      </teljes_nev>
23      <kor>50</kor>
24      <orvos_azonosito>2</orvos_azonosito>
25      <telefonszam>06307032140</telefonszam>
26      <beteg_azonositokod>2</beteg_azonositokod>
27      <bekerules_datuma>2022-10-19</bekerules_datuma>
28      <kezeloorvos>dr.Masik Ember</kezeloorvos>
29    </beteg>
30    <!--beteg 3-->
31    <beteg id="3">
32      <teljes_nev>
33        <vezeteknev>Kiss</vezeteknev>
34        <keresztnev>Tamas</keresztnev>
35      </teljes_nev>
36      <kor>33</kor>
37      <orvos_azonosito>3</orvos_azonosito>
38      <telefonszam>06307032142</telefonszam>
39      <beteg_azonositokod>3</beteg_azonositokod>
40      <bekerules_datuma>2022-12-19</bekerules_datuma>
41      <kezeloorvos>dr.Igazi Ember</kezeloorvos>
42    </beteg>
43  </betegnyilvantartas>
44 </betegnyilvantartorendszer>
```

```

</beteg>
<!--korhaz 1-->
<korhaz id="1">
  <cim>Valami utca 5</cim>
  <beosztottak>1</beosztottak>
  <hasznalt_gyogyszerek>algopirin</hasznalt_gyogyszerek>
  <osztalyok>Belgogyszer, Surgossegi</osztalyok>
  <gyogyszer_id>100</gyogyszer_id>
  <beteg_azonositokod>1</beteg_azonositokod>
  <bekerules_datuma>2022-11-19</bekerules_datuma>
  <mennyiseg>100</mennyiseg>
  <!--<munkahely>Surgossegi</munkahely>-->
</korhaz>
<!--korhaz 2-->
<korhaz id="2">
  <cim>Masik utca 5</cim>
  <beosztottak>2</beosztottak>
  <hasznalt_gyogyszerek>cataflam</hasznalt_gyogyszerek>
  <osztalyok>Belgogyszer, Surgossegi</osztalyok>
  <befogado_kepesseg>100</befogado_kepesseg>
  <beteg_azonositokod>3</beteg_azonositokod>
  <bekerules_datuma>2022-12-19</bekerules_datuma>
  <mennyiseg>100</mennyiseg>
  <!--<munkahely>Surgossegi</munkahely>-->
</korhaz>
<!--korhaz 3-->
<korhaz id="3">
  <cim>Azaz utca 5</cim>
  <beosztottak>3</beosztottak>
  <hasznalt_gyogyszerek>aspirin</hasznalt_gyogyszerek>
  <osztalyok>Belgogyszer, Surgossegi</osztalyok>
  <befogado_kepesseg>100</befogado_kepesseg>
  <beteg_azonositokod>2</beteg_azonositokod>
  <bekerules_datuma>2022-10-19</bekerules_datuma>
  <mennyiseg>100</mennyiseg>
  <!--<munkahely>Belgogyszer</munkahely>-->
</korhaz>
</korhaz>
<!--gyogyszer 1-->
<gyogyszer id="1">
  <ar>1000</ar>
  <gyogyszer_azonosito>1</gyogyszer_azonosito>
  <nev>algopirin</nev>
  <lejarat_ideje>2025-12-12</lejarat_ideje>
  <mennyiseg>12</mennyiseg>
</gyogyszer>
<!--gyogyszer 2-->
<gyogyszer id="2">
  <ar>1100</ar>
  <gyogyszer_azonosito>2</gyogyszer_azonosito>
  <nev>aspirin</nev>
  <lejarat_ideje>2025-12-12</lejarat_ideje>
  <mennyiseg>12</mennyiseg>
</gyogyszer>
<!--gyogyszer 3-->
<gyogyszer id="3">
  <ar>1500</ar>
  <gyogyszer_azonosito>3</gyogyszer_azonosito>
  <nev>cataflam</nev>
  <lejarat_ideje>2025-12-12</lejarat_ideje>
  <mennyiseg>12</mennyiseg>
</gyogyszer>

```

```

</gyogyszer>
<!--doktor 1-->
<doktor id="1">
  <teljes_nev>
    <vezeteknev>dr.Kitalalt</vezeteknev>
    <keresztnev>Ember</keresztnev>
  </teljes_nev>
  <telefonszam>111111111111</telefonszam>
  <szakterulet>Agysebesz</szakterulet>
  <doktor_azonositokod>1</doktor_azonositokod>
  <munkahely>Surgossegi</munkahely>
  <kezoorvos>dr.Kitalalt Ember</kezoorvos>
</doktor>
<!--doktor 2-->
<doktor id="2">
  <teljes_nev>
    <vezeteknev>dr.Masik</vezeteknev>
    <keresztnev>Ember</keresztnev>
  </teljes_nev>
  <telefonszam>222222222222</telefonszam>
  <szakterulet>Fogorvos</szakterulet>
  <doktor_azonositokod>2</doktor_azonositokod>
  <munkahely>Surgossegi</munkahely>
  <kezoorvos>dr.Masik Ember</kezoorvos>
</doktor>
<!--doktor 3-->
<doktor id="3">
  <teljes_nev>
    <vezeteknev>dr.Igazi</vezeteknev>
    <keresztnev>Ember</keresztnev>
  </teljes_nev>
  <telefonszam>333333333333</telefonszam>
  <szakterulet>Sziysebesz</szakterulet>
  <doktor_azonositokod>3</doktor_azonositokod>
  <munkahely>Belgyogyszat</munkahely>
  <kezoorvos>dr.Igazi Ember</kezoorvos>
</doktor>
</doktor>
<!--korkep beteg 1-->
<korkep id="1">
  <beteg_azonositokod>1</beteg_azonositokod>
  <datum>2022-11-19</datum>
  <diagnozis>Faj a feje</diagnozis>
  <gyogyszerek>algopirin</gyogyszerek>
  <kezo_orvos_neve>dr.Kitalalt Ember</kezo_orvos_neve>
</korkep>
<!--korkep beteg 2-->
<korkep id="2">
  <beteg_azonositokod>2</beteg_azonositokod>
  <datum>2022-10-19</datum>
  <diagnozis>Faj a feje</diagnozis>
  <gyogyszerek>cataflam</gyogyszerek>
  <kezo_orvos_neve>dr.Masik Ember</kezo_orvos_neve>
</korkep>
<!--korkep beteg 3-->
<korkep id="3">
  <beteg_azonositokod>3</beteg_azonositokod>
  <datum>2022-12-19</datum>
  <diagnozis>Faj a feje</diagnozis>
  <gyogyszerek>aspirin</gyogyszerek>
  <kezo_orvos_neve>dr.Igazi Ember</kezo_orvos_neve>
</korkep>
</betegnyilvantartas>
</betegnyilvantartorendszer>

```

1.d. XSD-schema leírás:

Az XML dokumentum tartalma alapján készített XSD-Schema tartalmaz a fő öt egyednél saját tulajdonságot és a Primary és Foreign Key-ek ábrázolásra

kerültek a dokumentumban. Minden element tartalmaz az elem által tartalmazott adatra vonatkozó típus megkötést.

1.d.1. XSD-schema:

```
XMLNXYCH1.xml  XMLSchemaNXYCH1.xsd  DOMReadNXYCH1.java  DOMQUERYNXYCH1.java  DomModifyNXYCH1.java
46      <element name="hasznalt_gyogyszerek" type="string"/></element>
47      <element name="osztalyok" type="string"/></element>
48      <element name="gyogyszer_id" type="integer"/></element>
49      <element name="beteg_azonositokod" type="integer"/>
50      <unique name="PKkbeteg_azonositokod">
51          <selector xpath="korhaz" />
52          <field xpath="beteg_azonositokod" />
53      </unique>
54  </element>
55  <element name="bekerules_datuma" type="date"/></element>
56  <element name="mennyiseg" type="integer"/></element>
57  <element name="munkahely" type="string"/></element>
58  </sequence>
59  <attribute name="id" type="string" />
60  </complexType>
61 </element>
62 <!-- <gyogyszerek adatai> -->
63 <element name="gyogyszer">
64   <complexType>
65     <sequence>
66       <element name="ar" type="integer"/></element>
67       <element name="gyogyszer_azonosito" type="integer">
68         <unique name="PKgyogyszer_azonosito">
69           <selector xpath="gyogyszer" />
70           <field xpath="gyogyszer_azonosito" />
71         </unique>
72       </element>
73       <element name="nev" type="string"/></element>
74       <element name="lejarat_ideje" type="date"/></element>
75       <element name="mennyiseg" type="integer"/></element>
76     </sequence>
77     <attribute name="id" type="string" />
78   </complexType>
79 </element>
80 <!-- <doktorok adatai> -->
81 <element name="doktor">
82   <complexType>
83     <sequence>
84       <element name="teljes_nev">
85         <complexType>
86           <sequence>
87             <element name="vezeteknev" type="string"/></element>
88             <element name="keresztnev" type="string"/></element>
89           </sequence>
90         </complexType>
91       </element>
92       <element name="telefonszam" type="integer"/></element>
93       <element name="szakterulet" type="string"/></element>
94       <element name="doktor_azonositokod" type="integer">
95         <unique name="PKdoktor_azonositokod">
```



```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <schema xmlns="http://www.w3.org/2001/XMLSchema"
3   targetNamespace="http://www.example.org/XMLSchemaNXYCH1"
4   xmlns:tns="http://www.example.org/XMLSchemaNXYCH1"
5   elementFormDefault="qualified">
6   <element name="betegnyilvantartorendszer">
7     <complexType>
8       <sequence>
9
10    <element name="betegnyilvantartas">
11      <complexType>
12        <sequence>
13          <!-- < beteg adatai > -->
14          <element name="beteg">
15            <complexType>
16              <sequence>
17                <complexType>
18                  <sequence>
19                    <element name="teljes_nev">
20                      <complexType>
21                        <sequence>
22                          <element name="vezeteknev" type="string"/></element>
23                          <element name="keresztnev" type="string"/></element>
24                        </sequence>
25                      </complexType>
26                    </element>
27                    <element name="kor" type="integer"/></element>
28                    <element name="orvos_azonosito" type="integer"/></element>
29                    <element name="beteg_azonositokod" type="integer">
30                      <unique name="PKbeteg_azonositokod">
31                        <selector xpath="beteg" />
32                        <field xpath="beteg_azonositokod" />
33                      </unique>
34                    </element>
35                    <element name="bekerules_datuma" type="date"/></element>
36                    <element name="kezelorvos" type="string"/></element>
37                  </sequence>
38                <attribute name="id" type="string" />
39              </complexType>
40            </element>
41            <!-- < korhazak adatai > -->
42            <element name="korhaz">
43              <complexType>
44                <sequence>
45                  <element name="cim" type="string"/></element>
46                  <element name="beosztottak" type="integer"/></element>
47                  <element name="hasznalt_gyogyszerek" type="string"/></element>
48                  <element name="osztalyok" type="string"/></element>
49                  <element name="gyogyszer_id" type="integer"/></element>
50                  <element name="beteg_azonositokod" type="integer">
51                    <unique name="PKbeteg_azonositokod">

```

```

81    <element name="doktor">
82      <complexType>
83        <sequence>
84          <element name="teljes_nev">
85            <complexType>
86              <sequence>
87                <element name="vezeteknev" type="string"/></element>
88                <element name="keresztnev" type="string"/></element>
89              </sequence>
90            </complexType>
91          </element>
92          <element name="telefonaszam" type="integer"/></element>
93          <element name="szakterulet" type="string"/></element>
94          <element name="doktor_azonositokod" type="integer">
95            <unique name="PKdoktor_azonositokod">
96              <selector xpath="doktor" />
97              <field xpath="doktor_azonositokod" />
98            </unique>
99          </element>
100         <element name="munkahely" type="string"/></element>
101         <element name="kezelorvos" type="string"/></element>
102       </sequence>
103       <attribute name="id" type="string" />
104     </complexType>
105   </element>
106   <!-- < korkepek adatai > -->
107   <element name="korkep">
108     <complexType>
109       <sequence>
110         <element name="beteg_azonositokod" type="integer">
111           <unique name="PKkbeteg_azonositokod">
112             <selector xpath="korkep" />
113             <field xpath="beteg_azonositokod" />
114           </unique>
115         </element>
116         <element name="datum" type="date"/></element>
117         <element name="diagnozis" type="string"/></element>
118         <element name="gyogyszerek" type="string"/></element>
119         <element name="kezo_orvos_neve" type="string"/></element>
120       </sequence>
121       <attribute name="id" type="string" />
122     </complexType>
123   </element>
124 </sequence>
125 </complexType>
126 </element>
127 </sequence>
128 </complexType>
129 </element>
130 </schema>

```

2. Feladat

2.1. DOMREAD leírása:

A DOMRead.java program felel a XML dokumentum teljes beolvasásáért. A program saját függvényeket használ a feladat későbbi programjaiban történő kód újra felhasználás miatt. A program a console-ra írja ki a beolvasott adatokat. Függvény készült az öt fő egyed csoport tartalmának kiírására.

2.1.1. DOMRead:

```
XMLNXYCH1.xml  XMLSchemaNXYCH1.xsd  DOMReadNXYCH1.java  DOMQUERYNXYCH1.java  DomModifyNXYCH1.java
1 package domcsomag;
2
3 import java.io.File;
4
5 import javax.xml.parsers.DocumentBuilder;
6 import javax.xml.parsers.DocumentBuilderFactory;
7
8 import org.w3c.dom.Document;
9 import org.w3c.dom.Element;
10 import org.w3c.dom.Node;
11 import org.w3c.dom.NodeList;
12
13 public class DOMReadNXYCH1 {
14
15     public static void main(String[] args) {
16         try {
17             // a beolvasás jelenleg egy metódus kód újrakezelt használata érdekében
18             betegkiiras();
19             korhazkiiras();
20             gyogyszerkiiras();
21             orvoskiiras();
22             korkepkiiras();
23         } catch (Exception e) {
24             System.out.println("A hiba: " + e);
25         }
26     }
27
28     public static void adatkiiras() {
29         try {
30             File xmlDocumentum = new File("XMLNXYCH1.xml");
31             DocumentBuilderFactory dFact = DocumentBuilderFactory.newInstance();
32             DocumentBuilder dBuild = dFact.newDocumentBuilder();
33             Document documentum = dBuild.parse(xmlDocumentum);
34
35             // kiolvassuk a root elementet documentum megkeresi a root megkapjuk a node
36             // nevét
37             System.out.println("Root element: " + documentum.getDocumentElement().getNodeName());
38
39             betegkiiras();
40             korhazkiiras();
41             gyogyszerkiiras();
42             orvoskiiras();
43             korkepkiiras();
44         } catch (Exception e) {
45             System.out.println("A hiba: " + e);
46         }
47     }
48
49     public static void betegkiiras() {
50         try {
51
```

```

49
50 public static void betegkiiras() {
51     try {
52         File xmlDocumentum = new File("XML\\NYCH1.xml");
53         DocumentBuilderFactory dFact = DocumentBuilderFactory.newInstance();
54         DocumentBuilder dBuild = dFact.newDocumentBuilder();
55         Document documentum = dBuild.parse(xmlDocumentum);
56
57         // kiolvassuk a root elementet documentum megkeresi a root megkapjuk a node
58         // nevét
59         System.out.println("Root element: " + documentum.getDocumentElement().getNodeName());
60
61         // a korhazak adatai a NodeList tömbből
62         System.out.println("---Korhazak kiolvasasa---");
63         NodeList betegList = documentum.getElementsByTagName("beteg");
64         for (int i = 0; i < betegList.getLength(); i++) {
65             Node betegNode = betegList.item(i);
66             System.out.println("Node neve " + betegNode.getNodeName() + " " + (i + 1));
67             if (betegNode.getNodeType() == Node.ELEMENT_NODE) {
68                 Element eElement = (Element) betegNode;
69                 System.out.println("Beteg teljes neve: " +
70                     eElement.getElementsByTagName("vezeteknev").item(0).getTextContent() + " " +
71                     eElement.getElementsByTagName("keresztnev").item(0).getTextContent());
72                 System.out.println("Beteg Kora: " + eElement.getElementsByTagName("kor").item(0).getTextContent());
73                 System.out.println("Beteg Orvosanak azonositoja: " + eElement.getElementsByTagName("orvos_azonosito").item(0).getTextContent());
74                 System.out.println("Beteg telefonszama: " + eElement.getElementsByTagName("telefonszam").item(0).getTextContent());
75                 System.out.println("Beteg azonositokodja: " + eElement.getElementsByTagName("beteg_azonositokod").item(0).getTextContent());
76                 System.out.println("Beteg bekerulesenek datuma: " + eElement.getElementsByTagName("bekerules_datuma").item(0).getTextContent());
77                 System.out.println("Beteg kezelorvos: " + eElement.getElementsByTagName("kezelorvos").item(0).getTextContent());
78
79                 System.out.println("-----");
80             }
81         }
82         System.out.println("---Ezek voltak a betegek---");
83
84     } catch (Exception e) {
85         System.out.println("A hiba: " + e);
86     }
87 }
88
89 public static void korhazkiiras() {
90     try {
91         File xmlDocumentum = new File("XML\\NYCH1.xml");
92         DocumentBuilderFactory dFact = DocumentBuilderFactory.newInstance();
93         DocumentBuilder dBuild = dFact.newDocumentBuilder();
94         Document documentum = dBuild.parse(xmlDocumentum);
95
96         // kiolvassuk a root elementet documentum megkeresi a root megkapjuk a node
97         // nevét
98         System.out.println("Root element: " + documentum.getDocumentElement().getNodeName());
99
100         // a korhazak adatai a NodeList tömbből
101         System.out.println("---Korhazak kiolvasasa---");
102         NodeList korhazList = documentum.getElementsByTagName("korhaz");
103         for (int i = 0; i < korhazList.getLength(); i++) {
104             Node korhazNode = korhazList.item(i);
105             System.out.println("Node neve " + korhazNode.getNodeName() + " " + (i + 1));
106             if (korhazNode.getNodeType() == Node.ELEMENT_NODE) {
107                 Element eElement = (Element) korhazNode;
108                 System.out.println("Korhaz cime: " + eElement.getElementsByTagName("cim").item(0).getTextContent());
109                 System.out.println("Korhaz beosztottai szama: "
110                     + eElement.getElementsByTagName("beosztottak").item(0).getTextContent());
111                 System.out.println("Korhazban hasznalt gyógyszerek: "
112                     + eElement.getElementsByTagName("hasznalt_gyogyszerek").item(0).getTextContent());
113                 System.out.println("Korhaz osztalyai: " + eElement.getElementsByTagName("osztalyok").item(0).getTextContent());
114                 System.out.println("Korhaz gyógyszer id: "
115                     + eElement.getElementsByTagName("gyogyszer_id").item(0).getTextContent());
116                 System.out.println("Korhaz beteg azonositoja: "
117                     + eElement.getElementsByTagName("beteg_azonositokod").item(0).getTextContent());
118                 System.out.println("Korhaz beteg bekerulesenek datuma: "
119                     + eElement.getElementsByTagName("bekerules_datuma").item(0).getTextContent());
120                 System.out.println("Korhaz gyógyszer mennyisege: "
121                     + eElement.getElementsByTagName("mennyiseg").item(0).getTextContent());
122                 // System.out.println("Korhazban levo orvos munkahelye: " +
123                 // eElement.getElementsByTagName("munkahely").item(0).getTextContent());
124                 System.out.println("-----");
125             }
126         }
127         System.out.println("---Ezek voltak a korhazak---");
128
129     } catch (Exception e) {
130         System.out.println("A hiba: " + e);
131     }
132 }
133

```

```

133     }
134
135     public static void gyogyszerkiiras() {
136         try {
137             File xmlDocumentum = new File("XMLNXYCH1.xml");
138             DocumentBuilderFactory dFact = DocumentBuilderFactory.newInstance();
139             DocumentBuilder dBuild = dFact.newDocumentBuilder();
140             Document documentum = dBuild.parse(xmlDocumentum);
141
142             // kiolvassuk a root elementet documentum megkeresi a root megkapjuk a node
143             // nevet
144             System.out.println("Root element: " + documentum.getDocumentElement().getNodeName());
145
146             // a gyogyszerek a NodeList tombbol
147             System.out.println("---Gyogyszerek kiolvasasa---");
148             NodeList gyoggyList = documentum.getElementsByTagName("gyogyszer");
149             for (int i = 0; i < gyoggyList.getLength(); i++) {
150                 Node gyoggyNode = gyoggyList.item(i);
151                 System.out.println("Node neve " + gyoggyNode.getNodeName() + " " + (i + 1));
152                 if (gyoggyNode.getNodeType() == Node.ELEMENT_NODE) {
153                     Element eElement = (Element) gyoggyNode;
154                     System.out
155                         .println("Gyogyszer ara: " + eElement.getElementsByTagName("ar").item(0).getTextContent());
156                     System.out.println("Gyogyszer azonositoja: "
157                         + eElement.getElementsByTagName("gyogyszerazonosito").item(0).getTextContent());
158                     System.out.println(
159                         "Gyogyszer neve: " + eElement.getElementsByTagName("nev").item(0).getTextContent());
160                     System.out.println("Gyogyszer lejarati datuma: "
161                         + eElement.getElementsByTagName("lejaratideje").item(0).getTextContent());
162                     System.out.println("Gyogyszerben levo mennyiseg: "
163                         + eElement.getElementsByTagName("mennyiseg").item(0).getTextContent());
164                     System.out.println("-----");
165                 }
166             }
167             System.out.println("---Ezek voltak a gyogyszerek---");
168
169         } catch (Exception e) {
170             System.out.println("A hiba: " + e);
171         }
172     }
173
174     public static void orvoskiiras() {
175         try {
176             File xmlDocumentum = new File("XMLNXYCH1.xml");
177             DocumentBuilderFactory dFact = DocumentBuilderFactory.newInstance();
178             DocumentBuilder dBuild = dFact.newDocumentBuilder();

```

```

172     }
173
174     public static void orvoskiiras() {
175         try {
176             File xmlDocumentum = new File("XMLNXYCH1.xml");
177             DocumentBuilderFactory dFact = DocumentBuilderFactory.newInstance();
178             DocumentBuilder dBuild = dFact.newDocumentBuilder();
179             Document documentum = dBuild.parse(xmlDocumentum);
180
181             // kiolvassuk a root elementet documentum megkeresi a root megkapjuk a node
182             // nevet
183             System.out.println("Root element: " + documentum.getDocumentElement().getNodeName());
184
185             // a orvosok adatai a NodeList tombbol
186             System.out.println("---Orvosok kiolvasasa---");
187             NodeList orvosList = documentum.getElementsByTagName("doktor");
188             for (int i = 0; i < orvosList.getLength(); i++) {
189                 Node orvosNode = orvosList.item(i);
190                 System.out.println("Node neve " + orvosNode.getNodeName() + " " + (i + 1));
191                 if (orvosNode.getNodeType() == Node.ELEMENT_NODE) {
192                     Element eElement = (Element) orvosNode;
193                     System.out.println(
194                         "Orvos teljes neve: " + eElement.getElementsByTagName("vezeteknev").item(0).getTextContent()
195                         + " " + eElement.getElementsByTagName("keresztnev").item(0).getTextContent());
196                     System.out.println("Orvos telefonszama: "
197                         + eElement.getElementsByTagName("telefonszam").item(0).getTextContent());
198                     System.out.println("Orvos szakterulete: "
199                         + eElement.getElementsByTagName("szakterulet").item(0).getTextContent());
200                     System.out.println("Orvos azonositokodja: "
201                         + eElement.getElementsByTagName("doktorazonositokod").item(0).getTextContent());
202                     System.out.println(
203                         "Orvos munkahelye: " + eElement.getElementsByTagName("munkahely").item(0).getTextContent());
204                     System.out.println("-----");
205                 }
206             }
207             System.out.println("---Ezek voltak az orvosok---");
208
209         } catch (Exception e) {
210             System.out.println("A hiba: " + e);
211         }
212     }
213

```



```

201         + element.getElementsByTagName( "doktor_azonositokod" ).item(0).getTextContent();
202         System.out.println(
203             "Orvos munkahelye: " + eElement.getElementsByTagName("munkahely").item(0).getTextContent();
204             System.out.println("-----");
205         }
206     }
207     System.out.println("---Ezek voltak az orvosok---");
208 }
209 } catch (Exception e) {
210     System.out.println("A hiba: " + e);
211 }
212 }
213 }
214 public static void korkepkiiras() {
215     try {
216         File xmlDocumentum = new File("XMLNXYCH1.xml");
217         DocumentBuilderFactory dFact = DocumentBuilderFactory.newInstance();
218         DocumentBuilder dBuild = dFact.newDocumentBuilder();
219         Document documentum = dBuild.parse(xmlDocumentum);
220
221         // kiolvassuk a root elementet documentum megkeresi a root megkapjuk a node
222         // nevét
223         System.out.println("Root element: " + documentum.getDocumentElement().getNodeName());
224
225         // a korkepek a NodeList tömbből
226         System.out.println("---Korkepek kiolvasasa---");
227         NodeList korkList = documentum.getElementsByTagName("korkep");
228         for (int i = 0; i < korkList.getLength(); i++) {
229             Node korkNode = korkList.item(i);
230             System.out.println("Node neve " + korkNode.getNodeName() + " " + (i + 1));
231             if (korkNode.getNodeType() == Node.ELEMENT_NODE) {
232                 Element eElement = (Element) korkNode;
233                 System.out.println("Beteg azonositokodja: "
234                     + eElement.getElementsByTagName("beteg_azonositokod").item(0).getTextContent();
235                 System.out.println(
236                     "Korkep datuma: " + eElement.getElementsByTagName("datum").item(0).getTextContent();
237                 System.out.println(
238                     "Diagnózis: " + eElement.getElementsByTagName("diagnozis").item(0).getTextContent();
239                 System.out.println(
240                     "Gyogyszerek: " + eElement.getElementsByTagName("gyogyszerek").item(0).getTextContent();
241                 System.out.println("Kezelo orvos neve: "
242                     + eElement.getElementsByTagName("kezelo_orvos_neve").item(0).getTextContent();
243                 System.out.println("-----");
244             }
245         }
246         System.out.println("---Ezek voltak a korkepek---");
247     } catch (Exception e) {
248         System.out.println("A hiba: " + e);
249     }
250 }
251 }
252 }

```

```

Problems  Javadoc  Declaration  Search  Console  Git Staging  Call Hierarchy
<terminated> DOMReadNXYCH1 [Java Application] C:\Users\Kiss Máté\p2\pool\plugins\org.eclipse.jst.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-1208\jre\bin\javaw.exe (2022. nov. 29. 18:42:34 - 18:42:35)

---Ezek voltak a gyogyszerek---
Root element: betegnyilvantartorendszer
---Orvosok kiolvasasa---
Node neve doktor1
Orvos teljes neve: dr.Kitalalt Ember
Orvos telefonszama: 1111111111111111
Orvos szakterulete: Agysebesz
Orvos azonositokodja: 1
Orvos munkahelye: Surgossegi
-----
Node neve doktor2
Orvos teljes neve: dr.Masik Ember
Orvos telefonszama: 2222222222222222
Orvos szakterulete: Fogorvos
Orvos azonositokodja: 2
Orvos munkahelye: Surgossegi
-----
Node neve doktor3
Orvos teljes neve: dr.Igazi Ember
Orvos telefonszama: 3333333333333333
Orvos szakterulete: Szivsebesz
Orvos azonositokodja: 3
Orvos munkahelye: Belgyogyszaszat
-----
---Ezek voltak az orvosok---
Root element: betegnyilvantartorendszer
---Korkepek kiolvasasa---
Node neve korkep1
Beteg azonositokodja: 1
Korkep datuma: 2022-11-19
Diagnózis: Faj a feje
Gyogyszerek: algopirin
Kezelo orvos neve: dr.Kitalalt Ember
-----
Node neve korkep2
Beteg azonositokodja: 2
Korkep datuma: 2022-10-19
Diagnózis: Faj a feje
Gyogyszerek: cataflam
Kezelo orvos neve: dr.Masik Ember
-----
Node neve korkep3
Beteg azonositokodja: 3
Korkep datuma: 2022-12-19
Diagnózis: Faj a feje
Gyogyszerek: aspirin
Kezelo orvos neve: dr.Igazi Ember
-----
---Ezek voltak a korkepek---

```

2.2. DOMQuery leírása:

A DOMQuery az adott öt főkategóriában lévő alkategóriáinak adatait tudja lekérdezni szintén saját függvényekkel. A program először bemutatja consolon milyen kategóriák és azon belül milyen alkategóriákra tudunk szűrni.

A minden opció is bevezetésre került különböző szinteken vagy a teljes XML dokumentum tartalmát olvassa be vagy csak az adott alkategóriákat, vagy az alkategórián belül az adott alcsoport tartalmát.

2.2.1. DOMQuery:

```
1 package domcsomag;
2
3 import java.io.File;
4
13 public class DOMQUERYNYCH1 {
14
15
16 public static void main(String[] args) {
17     try {
18         // documentum xml beolvasása hogy DOM parserrel kezeljem
19         File xmlDocumentum = new File("XMLNYCH1.xml");
20         DocumentBuilderFactory dFact = DocumentBuilderFactory.newInstance();
21         DocumentBuilder dBuild = dFact.newDocumentBuilder();
22         Document documentum = dBuild.parse(xmlDocumentum);
23
24         System.out.println("Lehetséges kategóriák amire szűrhet: ");
25         System.out.println("beteg");
26         System.out.println("korhaz");
27         System.out.println("gyogyszer");
28         System.out.println("doktor");
29         System.out.println("korkep");
30         System.out.println("minden");
31         System.out.println("Adja meg a lehetséges kategóriákból amire raak keresni");
32
33         // kategória beolvasása
34         Scanner input = new Scanner(System.in);
35         String opciok = input.next();
36         String categoria = opciok;
37         //kategóriák alkategóriáinak felsorolása és a minden adat kiírató opció implementálása
38         switch (categoria) {
39             case "beteg": {
40                 System.out.println(
41                     "A lehetséges keresési opciók: teljes_nev, kor, telefonszam, betegazonosítokod, bekerules_datuma, kezelo orvos, minden");
42                 System.out.println("ha most beírja hogy minden az adott kategóriából megkap mindent");
43                 String alkategoria = alkategoria();
44                 if (alkategoria.contains("minden")) {
45                     DOMReadNYCH1.betegkiiras();
46                 }
47                 break;
48             }
49             case "korhaz": {
50                 System.out.println(
51                     "A lehetséges keresési opciók: cim, beosztottak, hasznalt_gyogyszerek, osztalyok, befogado_kepesség, bekerules_datuma, mennyiség, minden");
52                 System.out.println("ha most beírja hogy minden az adott kategóriából megkap mindent");
53                 String alkategoria = alkategoria();
54                 if (alkategoria.contains("minden")) {
55                     DOMReadNYCH1.korhazkiiras();
56                 }
57                 break;
58             }
59             case "gyogyszer": {
60                 System.out.println(
61                     "A lehetséges keresési opciók: ar, gyogyszer_azonosito, nev, lejarat_ideje, mennyiség, minden");
62                 System.out.println("ha most beírja hogy minden az adott kategóriából megkap mindent");
63                 String alkategoria = alkategoria();
64                 if (alkategoria.contains("minden")) {
65                     DOMReadNYCH1.gyogyszerkiiras();
66                 }
67                 break;
68             }
69             case "doktor": {
70                 System.out.println(
71                     "A lehetséges keresési opciók: teljes_nev, telefonszam, szakterulet, doktor_azonosítokod, munkahely, minden");
72                 System.out.println("ha most beírja hogy minden az adott kategóriából megkap mindent");
73                 String alkategoria = alkategoria();
74                 if (alkategoria.contains("minden")) {
75                     DOMReadNYCH1.orvoskiiras();
76                 }
77                 break;
78             }
79             case "korkep": {
80                 System.out.println(
81                     "A lehetséges keresési opciók: beteg_azonosítokod, datum, diagnózis, gyogyszerek, kezelo_orvos_neve, minden");
82                 System.out.println("ha most beírja hogy minden az adott kategóriából megkap mindent");
83                 String alkategoria = alkategoria();
84                 if (alkategoria.contains("minden")) {
85                     DOMReadNYCH1.korkeпкиiras();
86                 }
87                 break;
88             }
89             case "minden": {
90                 System.out.println("Az összes adatot megkapja ha beírja hogy minden: ");
91                 DOMReadNYCH1.adatkiiras();
92                 break;
93             }
94         }
95
96         //a fő metódus ami a keresést végzi
97         behelyettesites();
98
99         System.out.println("----Ez volt a keresett adat----");
100
101     } catch (Exception e) {
102         System.out.println("A hiba: " + e);
103     }
104 }
```

```

XMLNXCH1.xml  XMLSchemaNXCH1.xsd  DOMReadNXCH1.java  DOMQUERYNXCH1.java  DomModifyNXCH1.java
95
96      //a fo metodus ami a keresest végzi
97      behelyettesites();
98
99      System.out.println("----Ez volt a keresett adat----");
100
101  } catch (Exception e) {
102      System.out.println("A hiba: " + e);
103  }
104
105
106  public static String fokategoria() {
107      // beolvasom a fokategoriat
108      Scanner scanner = new Scanner(System.in);
109      String adatok = scanner.next();
110      String fokategoria = adatok;
111      System.out.println("A választott fokategoria: " + fokategoria + "-----");
112      return fokategoria;
113  }
114
115  public static String alkategoria() {
116      // beolvasom az alkategoriat
117
118      Scanner scanner = new Scanner(System.in);
119      String adatok = scanner.next();
120      String alkategoria = adatok;
121      System.out.println("A választott alkategoria: " + alkategoria + "-----");
122      return alkategoria;
123  }
124
125  public static String behelyettesites() {
126      System.out.println("fokategoriat írja be újra:");
127      String tarolo = alkategoria();
128      String fokategoria = tarolo;
129
130      System.out.println("alkategoriat írja be újra:");
131      String tarolo2 = alkategoria();
132      String alkategoria = tarolo2;
133
134      try {
135
136          File xmlDocumentum = new File("XMLNXCH1.xml");
137          DocumentBuilderFactory dFact = DocumentBuilderFactory.newInstance();
138          DocumentBuilder dBuild = dFact.newDocumentBuilder();
139          Document documentum = dBuild.parse(xmlDocumentum);
140          // a teljes nev összetettséget kezelő ág
141          if (alkategoria.contains("teljes_nev")) {
142
143              System.out.println("----adatok kiolvasasa----");
144              NodeList nodeList = documentum.getElementsByTagName("beteg");
145              for (int i = 0; i < nodeList.getLength(); i++) {
146                  Node nNode = nodeList.item(i);
147
148                  System.out.println("Node neve " + nNode.getNodeName() + " " + (i + 1));
149                  if (nNode.getNodeType() == Node.ELEMENT_NODE) {
150                      Element eElement = (Element) nNode;
151                      System.out.println("Beteg teljes neve: "
152                          + eElement.getElementsByTagName("vezeteknev").item(0).getTextContent() + " "
153                          + eElement.getElementsByTagName("keresztnev").item(0).getTextContent());
154                      System.out.println("-----");
155                  }
156              }
157              //egyeb agak kezelése
158          } else if (alkategoria.contentEquals(alkategoria)) {
159
160              System.out.println("----Adatok kiolvasasa----");
161              NodeList nodeList = documentum.getElementsByTagName(fokategoria);
162              for (int i = 0; i < nodeList.getLength(); i++) {
163                  Node nNode = nodeList.item(i);
164                  System.out.println("Node neve " + nNode.getNodeName() + " " + (i + 1));
165                  if (nNode.getNodeType() == Node.ELEMENT_NODE) {
166                      Element eElement = (Element) nNode;
167                      System.out.println("Adatok: " + eElement.getElementsByTagName(alkategoria).item(0).getTextContent());
168                      System.out.println("-----");
169                  }
170              }
171          }
172      }
173  } catch (Exception e) {
174      System.out.println("A hiba: " + e);
175  }
176
177  }
178
179  return alkategoria;
180
181 }
182

```



```
XMLNXYCH1.xml  XMLSchemaNXYCH1.xsd  DOMReadNXYCH1.java  DOMQUERYNXYCH1.java  DomModifyNXYCH1.java
131  String tarolo2 = akategoria();
132  String alkategoria = tarolo2;
133

Problems  Javadoc  Declaration  Search  Console  Git Staging  Call Hierarchy
<terminated> DOMQUERYNXYCH1 [Java Application] C:\Users\Kiss Máté\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.2.v20220201-1208\jre\bin\javaw.exe (2022. nov. 29. 18:43:19 - 18:43:5
Lehetséges kategóriák amire szűrhet:
beteg
korhaz
gyogyszer
doktor
korkep
minden
Adja meg a lehetséges kategóriákból amire raakar keresni
beteg
A lehetséges keresési opciók: teljes_nev, kor, telefonszam, betegazonositokod, bekerules_datuma, kezeloorvos, minden
ha most beírja hogy minden az adott kategóriából megkap mindent
teljes_nev
A választott kategória: teljes_nev-----
fokategoriat írja be újra:
beteg
A választott kategória: beteg-----
alkategoriat írja be újra:
teljes_nev
A választott kategória: teljes_nev-----
---adatok kiolvasása---
Node neve beteg1
Beteg teljes neve: Kiss Mate
-----
Node neve beteg2
Beteg teljes neve: Kiss Antal
-----
Node neve beteg3
Beteg teljes neve: Kiss Tamas
-----
---Ez volt a keresett adat---
```

2.3. DOMModify leírása:

A DOMModify program beolvassa az XML dokumentumot és saját függvények bevezetésével az adott főcsoportokon belül képes a különböző id-vel rendelkező például beteg 1...3 specifikus adatának módosítására minden futás során egy specifikus adat kerül átírásra amit egy külön XML file-ba kerül mentésre ellenőrzés céljából. Minden adat átírható a függvények specifikus szűrése miatt.

2.3.1. DOMModify:

```
1 package domcsomag;
2
3 import org.w3c.dom.*;
4
5 public class DomModifyNXYCH1 {
6
7     private static final String File = "XMLNXYCH1.xml";
8     // private static final String UOFile = "test.xml";
9
10    public static void main(String[] args) {
11        NodeList list;
12
13        try {
14            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
15            DocumentBuilder builder = factory.newDocumentBuilder();
16
17            // fokategoria
18            // Adja meg a fokategoriat
19            System.out.println("Adja meg a fokategoriat");
20            String fokategoria = DOMQUERYNXYCH1.fokategoria();
21            // Alkategoria
22            // Adja meg az alkategoriat
23            System.out.println("Adja meg a alkategoriat");
24            String alkategoria = DOMQUERYNXYCH1.alkategoria();
25            // keresesi id megadasa
26            String idkod = idkereso();
27            // atirando eredeti ertekek
28            String adat = ertekhereso();
29            // ertekek amire atirunk
30            System.out.println("Adja meg mire akarja atirni az adatot");
31            String ujertekek = ertekvalto();
32
33            System.out.println("---Adatok kiolvasasa---");
34
35            // fajl beolvasasa
36            Document document = builder.parse(new File("XMLNXYCH1.xml"));
37            document.getDocumentElement().normalize();
38            // Aktuális elem meghatározása // "beteg"
39            list = document.getElementsByTagName(fokategoria);
40
41            for (int i = 0; i < list.getLength(); i++) {
42                Node node = list.item(i);
43                // Aktuális elem kiírása
44                System.out.println("\n Current element: " + node.getNodeName());
45                if (node.getNodeType() == Node.ELEMENT_NODE) {
46                    Element element = (Element) node;
47                    // id eltárolása egy stringbe // "id"
48                    String id = element.getAttribute("id");
49
50                    Node nodeid = list.item(i);
51                    if (nodeid.getNodeType() == Node.ELEMENT_NODE) {
52                        Element element1 = (Element) node;
```

```
66
67 // Aktuális elem kiírása
68 System.out.println("\n Current element: " + node.getNodeName());
69 if (node.getNodeType() == Node.ELEMENT_NODE) {
70     Element element = (Element) node;
71     // id eltárolása egy stringbe // "id"
72     String id = element.getAttribute("id");
73
74     Node nodeid = list.item(i);
75     if (nodeid.getNodeType() == Node.ELEMENT_NODE) {
76         Element element1 = (Element) node;
77         // id vizsgálata és ha megegyeznek a megadottal akkor ertekek modositasa
78         if (id.equals(idkod)) {
79             element1.getElementsByTagName(alkategoria).item(0).setTextContent(ujertekek);
80         }
81     }
82     // uj adatok kiirasa
83     System.out.println(
84         "\n Uj adat: " + element1.getElementsByTagName(alkategoria).item(0).getTextContent());
85 }
86 }
87 }
88 }
89 }
90
91 // Lekérdezi az adatokat és beírja egy fájlba
92 Transformer transformer = TransformerFactory.newInstance().newTransformer();
93 Source input = new DOMSource(document);
94 Result output = new StreamResult(new File("XMLNXYCH1UO.xml"));
95 transformer.transform(input, output);
96
97 } catch (ParserConfigurationException e) {
98     e.printStackTrace();
99 } catch (Exception e) {
100     e.printStackTrace();
101 }
102 }
103
104
105 public static String ertekkereso() {
106     // beolvasom az eredetiet
107     System.out.println("Adja meg az atirando adatot");
108     Scanner sc = new Scanner(System.in);
109     String adatok = sc.next();
110     String adat = adatok;
111     System.out.println("A atirando eredeti: " + adat + "-----");
112     return adat;
113 }
114
115 public static String tartalom() {
116     // beolvasom az eredeti alkategoria tartalmat
117     Scanner sc = new Scanner(System.in);
```

```

XMLNXYCH1.xml  XMLSchemaNXYCH1.xsd  DOMReadNXYCH1.java  DOMQUERYNXYCH1.java  DomModifyNXYCH1.java
93      Source input = new Unsource(document);
94      Result output = new StreamResult(new File("XMLNXYCH1U3.xml"));
95      transformer.transform(input, output);
96
97      } catch (ParserConfigurationException e) {
98          e.printStackTrace();
99      } catch (Exception e) {
100          e.printStackTrace();
101      }
102  }
103  }
104
105  public static String ertekkereso() {
106      // beolvasom az eredeti
107      System.out.println("Adja meg az atirando adatot");
108      Scanner sc = new Scanner(System.in);
109      String adatok = sc.next();
110      String adat = adatok;
111      System.out.println("A atirando eredeti: " + adat + "-----");
112      return adat;
113  }
114
115  public static String tartalom() {
116      // beolvasom az eredeti alkategoria tartalmat
117      Scanner sc = new Scanner(System.in);
118      String adatok = sc.next();
119      String tartalom = adatok;
120      System.out.println("Eredeti tartalom: " + tartalom + "-----");
121      return tartalom;
122  }
123
124      // uj erteket ad meg
125  public static String ertekvalto() {
126      System.out.println("Adja meg az uj erteket");
127      Scanner sc = new Scanner(System.in);
128      String adatok = sc.next();
129      String ujertek = adatok;
130      System.out.println("Az uj ertek: " + ujertek + "-----");
131      return ujertek;
132  }
133
134      // id alapu kereses
135  public static String idkereso() {
136      System.out.println("Adja meg az elem idt");
137      Scanner sc = new Scanner(System.in);
138      String adatok = sc.next();
139      String id = adatok;
140      return id;
141  }
142  }
143  }
144
Problems  Javadoc  Declaration  Search  Console  Git Staging  Call Hierarchy
<terminated> DOMQUERYNXYCH1 [Java Application] C:\Users\Kiss Máté\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-1208\jre\bin\javaw.exe (2022. nov. 29. 18:43:19 - 18:43:50)
Lehetseges kategoriak amire szurhet:

```

Figyelem a 2.beteg életkora 50 volt! új értéke 90!

```

Problems  Javadoc  Declaration  Search  Console  ×
<terminated> DomModifyNXYCH1 [Java Application] C:\Users\Kiss Máté
Adja meg a fokategoriat
beteg
A választott fokategoria: beteg-----
Adja meg a alkategoriat
kor
A választott alkategoria: kor-----
Adja meg az elem idt
2
Adja meg az atirando adatot
50
A atirando eredeti: 50-----
Adja meg mire akarja atirni az adatot
Adja meg az uj erteket
90
Az uj ertek: 90-----
---Adatok kiolvasasa---

Current element: beteg

Uj adat: 23

Current element: beteg

Uj adat: 90

Current element: beteg

Uj adat: 33

```