

# JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Orvosok

Készítette: **Juhász Marcell Tibor**

Neptunkód: **O9V4M0**

Dátum: **2022.11.27.**

# 1)Feladat leírása:

A beadandó feladatom témája az előző munkahelyem ihlette, ahol magánklinikának a működéséhez szükséges adatoknak készítettem strukturált adattárolást.

Ez egy olyan adatbázis, amelyben megtalálható az orvosoknak a részletes adatai, szakrendelések adatai majd az ezekből összeállított határidő napló. A naplóban vezetve vannak az elő jegyzett időpontok, amelyekben szerepelnek a páciens adatai, beavatkozás részletei, például melyik orvos, melyik teremben és mikor hajtotta végre az adott szakrendelést.

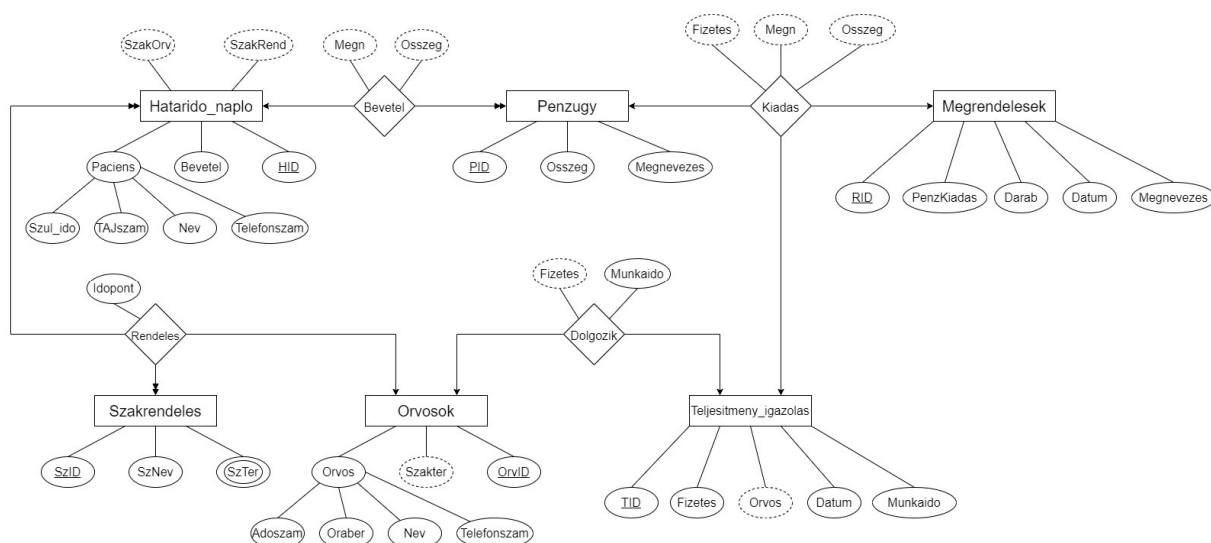
A klinika pénzügyeire is kitérek, amelyekben a kiadás vagy bevételről szóló bejegyzések találhatóak.

Az orvosok munkájáról is ejtek bejegyzéseket a teljesítmény igazolásokban, amelyekben az orvoshoz tartozó fizetések, óraszámok találhatóak.

A megrendeléseket is nyilvántartom amelyek egy összeg, és megnevezés tulajdonságokat kaptak.

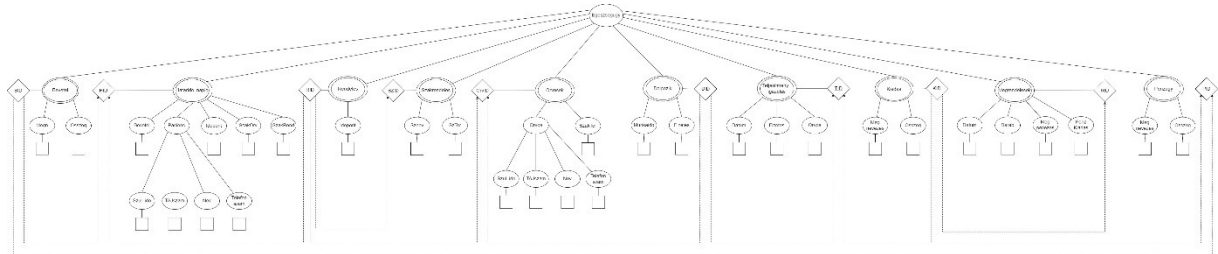
## 1a)ER modell:

Az ER modell a magánklinikának készített adatbázis felépítését mutatja be. Rendelkezik 1:1, 1:N, N:M kapcsolatokkal. A kapcsolat fajtái a nyilakkal különböztetem meg.



### 1b) XDM modell:

Az ER modell alapján készítettem, amellyel szeretném a későbbi XML fájl felépítését szemléltetni.



### 1c)XML:

Az XMD modell alapján kezdtem neki az XML fájl kiépítésének. Mindenhol próbáltam legalább 3 egyedet létrehozni, a későbbi feladatok könnyebb elemezhetőségeért. A kapcsolatokat is létrehoztam az ER modellben szereplőiekhez hasonlóan.

Először kezdtem a bevételekkel:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <egeszsegugy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:noNamespaceSchemaLocation="XMLSchema09V4M0.xml">
5   <bevetel BID="BID01" HID="HID01">
6     <megnevezes>CT</megnevezes>
7     <osszeg>20000</osszeg>
8   </bevetel>
9   <bevetel BID="BID02" HID="HID01">
10     <megnevezes>Ultrahang</megnevezes>
11     <osszeg>22000</osszeg>
12   </bevetel>
13   <bevetel BID="BID03" HID="HID01">
14     <megnevezes>Mammográfia</megnevezes>
15     <osszeg>15780</osszeg>
16   </bevetel>
```

Aztán folytattam a határidő naplókkel:

```
1
2   <hatarido_naplo HID="HID01">
3     <bevetel>20000</bevetel>
4     <paciens>
5       <nev>Kiss Gyurkó</nev>
6       <TAJszam>5684138744532</TAJszam>
7       <Szul_ido>1985.05.06.</Szul_ido>
8       <telefonszam>36208541254</telefonszam>
9     </paciens>
10    <idopont>2022.08.12.</idopont>
11    <szakorv>1</szakorv>
12    <szakrend>1</szakrend>
13  </hatarido_naplo>
14  <hatarido_naplo HID="HID02">
15    <bevetel>22000</bevetel>
16    <paciens>
17      <nev>Szakosné Dorina</nev>
18      <TAJszam>84571218574</TAJszam>
19      <Szul_ido>1993.02.08.</Szul_ido>
20      <telefonszam>3670854128</telefonszam>
21    </paciens>
22    <idopont>2022.08.15.</idopont>
23    <szakorv>2</szakorv>
24    <szakrend>2</szakrend>
25  </hatarido_naplo>
26  <hatarido_naplo HID="HID03">
27    <bevetel>15780</bevetel>
28    <paciens>
29      <nev>Ferencné Marianna</nev>
30      <TAJszam>23654178</TAJszam>
31      <Szul_ido>1958.12.31.</Szul_ido>
32      <telefonszam>3630541254</telefonszam>
33    </paciens>
34    <idopont>2022.09.01.</idopont>
35    <szakorv>3</szakorv>
36    <szakrend>3</szakrend>
37  </hatarido_naplo>
38  <hatarido_naplo HID="HID04">
39    <bevetel>35600</bevetel>
40    <paciens>
41      <nev>Kiss Gézáné</nev>
42      <TAJszam>47854123</TAJszam>
43      <Szul_ido>1962.03.24.</Szul_ido>
44      <telefonszam>3620854125</telefonszam>
45    </paciens>
46    <idopont>2022.09.02.</idopont>
47    <szakorv>3</szakorv>
48    <szakrend>3</szakrend>
49  </hatarido_naplo>
50
```

Ezután következtek a rendelések kapcsolat:

```
1   <rendeles RID="RID01" HID="HID01">
2     <idopont>2022.08.12.</idopont>
3   </rendeles>
4   <rendeles RID="RID02" HID="HID02">
5     <idopont>2022.08.15.</idopont>
6   </rendeles>
7   <rendeles RID="RID03" HID="HID03">
8     <idopont>2022.09.01.</idopont>
9   </rendeles>
10  <rendeles RID="RID04" HID="HID04">
11    <idopont>2022.09.02.</idopont>
12  </rendeles>
13
```

Majd utána a szakrendeléseket soroltam fel:

```
1 <szakrendeles SZID="SZID01">
2   <Sznev>CT</Sznev>
3   <Szter>1</Szter>
4 </szakrendeles>
5 <szakrendeles SZID="SZID02">
6   <Sznev>Ultrahang</Sznev>
7   <Szter>2</Szter>
8 </szakrendeles>
9 <szakrendeles SZID="SZID03">
10  <Sznev>Mammográfia</Sznev>
11  <Szter>3</Szter>
12 </szakrendeles>
13
```

Mindezek után az orvosok adatait vittem fel:

```
1 <orvosok OrvID="ORV1">
2   <orvos>
3     <nev>Dr. Nagy Béla</nev>
4     <Szul_ido>1958.03.12.</Szul_ido>
5     <TAJszam>123456789</TAJszam>
6     <Telefonszam>36504125412</Telefonszam>
7   </orvos>
8   <szakter>1</szakter>
9 </orvosok>
10 <orvosok OrvID="ORV2">
11   <orvos>
12     <nev>Dr. Kiss Ferencné</nev>
13     <Szul_ido>1976.11.23.</Szul_ido>
14     <TAJszam>52416387</TAJszam>
15     <Telefonszam>36205244156</Telefonszam>
16   </orvos>
17   <szakter>2</szakter>
18 </orvosok>
19 <orvosok OrvID="ORV3">
20   <orvos>
21     <nev>Dr. Kiss Ferenc</nev>
22     <Szul_ido>1974.10.21.</Szul_ido>
23     <TAJszam>976431824</TAJszam>
24     <Telefonszam>36305426457</Telefonszam>
25   </orvos>
26   <szakter>3</szakter>
27 </orvosok>
28
```

Utána a dolgozik kapcsolatot definiáltam:

```
1 <dolgozik DID="DID01" OrvID="ORV1">
2   <munkaido>8</munkaido>
3   <fizetes>56000</fizetes>
4 </dolgozik>
5 <dolgozik DID="DID02" OrvID="ORV2">
6   <munkaido>4</munkaido>
7   <fizetes>65300</fizetes>
8 </dolgozik>
9 <dolgozik DID="DID03" OrvID="ORV3">
10  <munkaido>12</munkaido>
11  <fizetes>98000</fizetes>
12 </dolgozik>
13
```

Ezután a teljesítmény igazolásokat hoztam létre:

```
1 <teljesitmeny_igazolas TID="TID01">
2   <datum>2022.08.12.</datum>
3   <fizetes>56000</fizetes>
4   <orvos>1</orvos>
5 </teljesitmeny_igazolas>
6 <teljesitmeny_igazolas TID="TID02">
7   <datum>2022.08.15.</datum>
8   <fizetes>65300</fizetes>
9   <orvos>2</orvos>
10 </teljesitmeny_igazolas>
11 <teljesitmeny_igazolas TID="TID03">
12   <datum>2022.08.12.</datum>
13   <fizetes>98000</fizetes>
14   <orvos>3</orvos>
15 </teljesitmeny_igazolas>
16
```

Majd a kiadás kapcsolatot hoztam létre:

```
1 <kiadas KID="KID01" TID="TID01">
2   <megnevezes>orvosbér</megnevezes>
3   <osszeg>56000</osszeg>
4 </kiadas>
5 <kiadas KID="KID02" TID="TID02">
6   <megnevezes>orvosbér</megnevezes>
7   <osszeg>65300</osszeg>
8 </kiadas>
9 <kiadas KID="KID03" TID="TID03">
10   <megnevezes>orvosbér</megnevezes>
11   <osszeg>98000</osszeg>
12 </kiadas>
13 <kiadas KID="KID04" RID="RID01">
14   <megnevezes>felszerelés</megnevezes>
15   <osszeg>12000</osszeg>
16 </kiadas>
17 <kiadas KID="KID05" RID="RID02">
18   <megnevezes>ételrendelés</megnevezes>
19   <osszeg>35000</osszeg>
20 </kiadas>
21
```

Ezek után a megrendelésekhez generáltam adatokat:

```
1 <megrendelesek RID="RID01">
2   <datum>2022.09.05.</datum>
3   <darab>1</darab>
4   <megnevezes>Gumikesztyű</megnevezes>
5   <penzkiadas>12000</penzkiadas>
6 </megrendelesek>
7 <megrendelesek RID="RID02">
8   <datum>2022.09.06.</datum>
9   <darab>4</darab>
10   <megnevezes>Pizza</megnevezes>
11   <penzkiadas>35000</penzkiadas>
12 </megrendelesek>
13
```

Végezetül a pénzügyet hoztam létre.

```
1 <penzugy PID="PID01">
2   <megnevezes>TID01</megnevezes>
3   <osszeg>56000</osszeg>
4 </penzugy>
5 <penzugy PID="HID02">
6   <megnevezes>Bevétel</megnevezes>
7   <osszeg>22000</osszeg>
8 </penzugy>
9 </egeszsegugy>
```

## 1d)XMLSchema:

A létrehozott XML alapján indultam neki a séma létrehozásának. Ebben definiálom az XML-ben található elemeket. Saját típusokat nem definiáltam a későbbi egyszerűbb validáció miatt.

Először a bevételhez tartozó stílust hozom létre:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <xs:schema attributeFormDefault="unqualified" elementFormDefault=
  "qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
3   <xs:element name="egeszsegugy">
4     <xs:complexType>
5       <xs:sequence>
6         <xs:element maxOccurs="unbounded" name="bevetel">
7           <xs:complexType>
8             <xs:sequence>
9               <xs:element name="megnevezes" type="xs:string" />
10              <xs:element name="osszeg" type="xs:unsignedShort" />
11            </xs:sequence>
12            <xs:attribute name="BID" type="xs:string" use="required" />
13          </xs:complexType>
14        </xs:element>
15      </xs:sequence>
16    </xs:complexType>
17  </xs:element>
18 </xs:schema>
```

Másodiknak a határidőnaplóhoz tartozót:

```
1      <xs:element maxOccurs="unbounded" name="hatarido_naplo">
2          <xs:complexType>
3              <xs:sequence>
4                  <xs:element name="bevetel" type="xs:unsignedShort" />
5                  <xs:element name="paciens">
6                      <xs:complexType>
7                          <xs:sequence>
8                              <xs:element name="nev" type="xs:string" />
9                              <xs:element name="TAJszam" type="xs:unsignedLong" />
10                             <xs:element name="Szul_ido" type="xs:string" />
11                             <xs:element name="telefonszam" type="
12                                 "xs:unsignedLong" />
13                         </xs:sequence>
14                     </xs:complexType>
15                 </xs:element>
16                 <xs:element name="idopont" type="xs:string" />
17                 <xs:element name="szakorv" type="xs:unsignedByte" />
18                 <xs:element name="szakrend" type="xs:unsignedByte" />
19             </xs:sequence>
20             <xs:attribute name="HID" type="xs:string" use="required" />
21         </xs:complexType>
22     </xs:element>
23
```

Következőnek a rendelésnek hozom létre:

```
1      <xs:element maxOccurs="unbounded" name="rendeles">
2          <xs:complexType>
3              <xs:sequence>
4                  <xs:element name="idopont" type="xs:string" />
5              </xs:sequence>
6              <xs:attribute name="RID" type="xs:string" use="required" />
7          </xs:complexType>
8      </xs:element>
9
```



Negyediknek a szakrendelését hozom létre:

```
1      <xs:element maxOccurs="unbounded" name="szakrendeles">
2          <xs:complexType>
3              <xs:sequence>
4                  <xs:element name="Sznev" type="xs:string" />
5                  <xs:element name="Szter" type="xs:unsignedByte" />
6              </xs:sequence>
7              <xs:attribute name="SZID" type="xs:string" use="required" />
8          </xs:complexType>
9      </xs:element>
10
```

Ezután az orvosokhoz tartozó stílust hozom létre:

```
1      <xs:element maxOccurs="unbounded" name="orvosok">
2          <xs:complexType>
3              <xs:sequence>
4                  <xs:element name="orvos">
5                      <xs:complexType>
6                          <xs:sequence>
7                              <xs:element name="nev" type="xs:string" />
8                              <xs:element name="Szul_ido" type="xs:string" />
9                              <xs:element name="TAJszam" type="xs:unsignedInt" />
10                             <xs:element name="Telefonszam" type="
11                                 "xs:unsignedLong" />
12                             </xs:sequence>
13                         </xs:complexType>
14                     </xs:element>
15                     <xs:element name="szakter" type="xs:unsignedByte" />
16                 </xs:sequence>
17                 <xs:attribute name="OrvID" type="xs:string" use="required"
18             />
19         </xs:complexType>
20     </xs:element>
21
```

Ezek után a dolgozik kapcsolatét hozom létre:

```
1      <xs:element maxOccurs="unbounded" name="dolgozik">
2        <xs:complexType>
3          <xs:sequence>
4            <xs:element name="munkaido" type="xs:unsignedByte" />
5            <xs:element name="fizetes" type="xs:unsignedInt" />
6          </xs:sequence>
7          <xs:attribute name="DID" type="xs:string" use="required" />
8        </xs:complexType>
9      </xs:element>
10
```

Hetediknek a teljesítmény igazolását hozom létre:

```
1      <xs:element maxOccurs="unbounded" name="teljesitmeny_igazolas">
2        <xs:complexType>
3          <xs:sequence>
4            <xs:element name="datum" type="xs:string" />
5            <xs:element name="fizetes" type="xs:unsignedInt" />
6            <xs:element name="orvos" type="xs:unsignedByte" />
7          </xs:sequence>
8          <xs:attribute name="TID" type="xs:string" use="required" />
9        </xs:complexType>
10     </xs:element>
11
```

Utána a kiadásét hozom létre:

```
1      <xs:element maxOccurs="unbounded" name="kiadas">
2        <xs:complexType>
3          <xs:sequence>
4            <xs:element name="megnevezes" type="xs:string" />
5            <xs:element name="osszeg" type="xs:unsignedInt" />
6          </xs:sequence>
7          <xs:attribute name="KID" type="xs:string" use="required" />
8        </xs:complexType>
9      </xs:element>
10
```

Utolsó előtt a megrendelését hozom létre:

```
1      <xs:element maxOccurs="unbounded" name="megrendelesek">
2          <xs:complexType>
3              <xs:sequence>
4                  <xs:element name="datum" type="xs:string" />
5                  <xs:element name="darab" type="xs:unsignedByte" />
6                  <xs:element name="megnevezes" type="xs:string" />
7                  <xs:element name="penzkiadas" type="xs:unsignedShort" />
8              </xs:sequence>
9              <xs:attribute name="RID" type="xs:string" use="required" />
10         </xs:complexType>
11     </xs:element>
12
```

Utoljára a pénzügyét hozom létre:

```
1      <xs:element maxOccurs="unbounded" name="penzugy">
2          <xs:complexType>
3              <xs:sequence>
4                  <xs:element name="megnevezes" type="xs:string" />
5                  <xs:element name="osszeg" type="xs:unsignedShort" />
6              </xs:sequence>
7              <xs:attribute name="PID" type="xs:string" use="required" />
8          </xs:complexType>
9      </xs:element>
10  </xs:sequence>
11  </xs:complexType>
12  </xs:element>
13 </xs:schema>
```

## 2.Feladat

### 2a)DomRead:

Az XML fájlból a megadott útvonalakon különféle szűréseknek megfelelően lekérdezéseket hajt végre. Ezeket egy TXT fájlba kiírja. A sok import után kezdem a képeket.

```
1 public class DOMRead09V4M0 {
2
3     private static Document doc;
4
5     public static void main(String argv[]) throws SAXException, IOException, ParserConfigurationException {
6         File xmlFile = new File("./XML09V4M0.xml");
7
8         DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
9         DocumentBuilder dBuilder = factory.newDocumentBuilder();
10        doc = dBuilder.parse(xmlFile);
11
12        doc.getDocumentElement().normalize();
13
14        System.out.println("Root element: " + doc.getDocumentElement().getNodeName());
15        //Az osszes megrendeles kiirasa.
16        readAllMegrendeles();
17
18        //Az osszes kiadas kiirasa.
19        readAllKiadas();
20
21        //Hatirido naplok kiirasa.
22        readAllHatarido_naplo();
23
24        //Orvosok listajanak kiirasa.
25        readAllOrvos();
26    }
27
```

```
1 private static void readAllMegrendeles() {
2     NodeList megrendesekekList = doc.getElementsByTagName("megrendesekek");
3     for (int i = 0; i < megrendesekekList.getLength(); i++) {
4
5         Node nNode = megrendesekekList.item(i);
6
7         if (nNode.getNodeType() == Node.ELEMENT_NODE) {
8
9             Element element = (Element) nNode;
10
11             String rid = element.getAttribute("RID");
12
13             //Fajlba iras
14             try (FileWriter fw = new FileWriter("09V4M0DOMRead.txt", true);) {
15                 fw.write("Azonosito: "+ rid +"\n");
16                 fw.write("Megnevezes: "+ element.getElementsByTagName("megnevezes").item(0).getTextContent() +"\n");
17                 fw.write("Kiadas: "+element.getElementsByTagName("penzkiadas").item(0).getTextContent()+ "\n");
18                 fw.write("Darab: "+element.getElementsByTagName("darab").item(0).getTextContent()+ "\n\n");
19             } catch (IOException e) {
20                 e.printStackTrace();
21             }
22         }
23     }
24 }
25
26
```

```

1 private static void readAllKiadas() {
2     NodeList kiadasokList = doc.getElementsByTagName("kiadas");
3     for (int i = 0; i < kiadasokList.getLength(); i++) {
4
5         Node nNode = kiadasokList.item(i);
6
7         if (nNode.getNodeType() == Node.ELEMENT_NODE) {
8
9             Element element = (Element) nNode;
10
11             String kid = element.getAttribute("KID");
12             String secondId = "";
13
14             if(element.getAttribute("TID") != null) {
15                 secondId = element.getAttribute("TID");
16             }else if(element.getAttribute("RID") != null) {
17                 secondId = element.getAttribute("RID");
18             }
19
20             //Fajba iras
21             try (FileWriter fw = new FileWriter("09V4M0DOMRead.txt", true);) {
22                 fw.write("Azonosito: " + kid + " " + secondId + "\n");
23                 fw.write("Megnevezes: " + element.getElementsByTagName("megnevezes").item(0).getTextContent() + "\n");
24                 fw.write("Osszeg: " + element.getElementsByTagName("osszeg").item(0).getTextContent() + "\n\n");
25             } catch (IOException e) {
26                 e.printStackTrace();
27             }
28         }
29     }
30 }
31 }
32

```

```

1 private static void readAllHatarido_naplo() {
2     NodeList hataridolist = doc.getElementsByTagName("hatarido_naplo");
3     for (int i = 0; i < hataridolist.getLength(); i++) {
4
5         Node nNode = hataridolist.item(i);
6
7         if (nNode.getNodeType() == Node.ELEMENT_NODE) {
8
9             Element element = (Element) nNode;
10
11             String hid = element.getAttribute("HID");
12
13             //Fajba iras
14             try (FileWriter fw = new FileWriter("09V4M0DOMRead.txt", true);) {
15                 Node paciens = (Node) element.getElementsByTagName("paciens");
16                 fw.write("Azonosito: " + hid + "\n");
17                 fw.write("Idopont: " + element.getElementsByTagName("ido pont").item(0).getTextContent() + "\n");
18                 fw.write("Bevetel: " + element.getElementsByTagName("bevetel").item(0).getTextContent() + "\n");
19                 fw.write("Szakorvos: " + element.getElementsByTagName("szakorv").item(0).getTextContent() + "\n");
20                 fw.write("Szakrendelo: " + element.getElementsByTagName("szakrend").item(0).getTextContent() + "\n");
21                 fw.write("Paciens neve: " + paciens.getAttributes().getNamedItem("nev").getTextContent() + "\n");
22                 fw.write("Paciens TAJszama: " + paciens.getAttributes().getNamedItem("TAJszam").getTextContent() + "\n");
23                 fw.write("Paciens telefonszama: " + paciens.getAttributes().getNamedItem("telefonszam").getTextContent() + "\n");
24                 fw.write("Paciens szuletesi ideje: " + element.getElementsByTagName("Szul_ido").item(0).getTextContent() + "\n\n");
25             } catch (IOException e) {
26                 e.printStackTrace();
27             }
28         }
29     }
30 }
31

```

```

1 private static void readAllOrvos(){
2     NodeList orvosokList = doc.getElementsByTagName("orvosok");
3     for (int i = 0; i < orvosokList.getLength(); i++) {
4
5         Node nNode = orvosokList.item(i);
6
7         if (nNode.getNodeType() == Node.ELEMENT_NODE) {
8
9             Element element = (Element) nNode;
10
11             String orvID = element.getAttribute("OrvID");
12
13             //Fajlba iras
14             try (FileWriter fw = new FileWriter("09V4M0DOMRead.txt", true);) {
15                 Node orvos = (Node) element.getElementsByTagName("orvos");
16                 fw.write("Azonosito: " + orvID + "\n");
17                 fw.write("Nev: " + orvos.getAttributes().getNamedItem("nev").getTextContent() + "\n");
18                 fw.write("Szuletesi ido: " + orvos.getAttributes().getNamedItem("Szul_ido").getTextContent() + "\n");
19                 fw.write("TAJ szam: " + orvos.getAttributes().getNamedItem("TAJszam").getTextContent() + "\n");
20                 fw.write("Telefonszam: " + orvos.getAttributes().getNamedItem("Telefonszam").getTextContent() + "\n");
21                 fw.write("Szakterulet: " + element.getElementsByTagName("szakter").item(0).getTextContent() + "\n\n");
22             } catch (IOException e) {
23                 e.printStackTrace();
24             }
25         }
26     }
27 }
28

```

## 2a)DomModify:

Az XML fájlból beolvasott elementeken hajtok végre manipulációkat, amelyet egy külön módosított XML fájlban eltárolok. A képeket az importok után kezdem.



```

1 public class DOMModify09V4M0 {
2     public static void main(String argv[]) throws SAXException, IOException, ParserConfigurationException {
3         File xmlFile = new File("./XML09V4M0.xml");
4
5         DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
6         DocumentBuilder dBuilder = factory.newDocumentBuilder();
7         Document doc = dBuilder.parse(xmlFile);
8
9         //A masodik hatarido naplo bevetelenek modosotisa.
10        NodeList hataridoNaplok = doc.getElementsByTagName("hatarido_naplo");
11        Node naplo = hataridoNaplok.item(1);
12        naplo.getAttributes().getNamedItem("bevetel").setTextContent("30000");
13
14        //Kiadások összegének AFA-s novelese.
15        NodeList kiadasok = doc.getElementsByTagName("kiadas");
16        for(int i = 0; i < kiadasok.getLength(); i++) {
17            Node kiadas = kiadasok.item(i);
18            int osszeg = Integer.parseInt(kiadas.getAttributes().getNamedItem("osszeg").toString());
19            String osszegString = "" + osszeg * 1.27;
20            kiadas.getAttributes().getNamedItem("osszeg").setTextContent(osszegString);
21        }
22
23        //Az orvosok fizetesének emelese 32%-al.
24        NodeList orvosokFizetesLista = doc.getElementsByTagName("dolgozik");
25        multiplyListElement(orvosokFizetesLista, "fizetes", 1.32);
26
27        //Megrendelesek darab elementjének torlese
28        NodeList megrendelesekLista = doc.getElementsByTagName("megrendelesek");
29        for (int i = 0; i < megrendelesekLista.getLength(); i++) {
30            Node megrendeles = megrendelesekLista.item(i);
31            NodeList childNodes = megrendeles.getChildNodes();
32            for (int j = 0; j < childNodes.getLength(); j++) {
33                Node item = childNodes.item(j);
34                if (item.getNodeType() == Node.ELEMENT_NODE) {
35                    if ("darab".equalsIgnoreCase(item.getNodeName())) {
36                        megrendeles.removeChild(item);
37                    }
38                }
39            }
40        }
41    }
42
43    //Fajl mentese
44    try (FileOutputStream output = new FileOutputStream("09V4M0modified.xml"))
45    {
46        writeXml(doc, output);
47    } catch (TransformerException e)
48    {
49        e.printStackTrace();
50    }
51 }
52

```

```

1     private static void multiplyListElement(NodeList nodeList, String tagName, double percentageValue) {
2         if (nodeList != null) {
3             for (int i = 0; i < nodeList.getLength(); i++) {
4                 Node node = nodeList.item(i);
5                 int newVal = Integer.parseInt(node.getAttributes().getNamedItem(tagName).toString());
6                 String newValString = "" + newVal * percentageValue;
7                 Element elem = (Element) node;
8                 try {
9                     elem.getAttributes().getNamedItem(tagName).setTextContent(newValString);
10                } catch (DOMException e) {
11                    e.printStackTrace();
12                }
13            }
14        }
15    }
16

```

```
1 //A módosított XML fájlba írása.
2 private static void writeXml(Document doc, OutputStream output) throws TransformerException,
  UnsupportedEncodingException {
3
4     TransformerFactory transformerFactory = TransformerFactory.newInstance();
5
6     Transformer transformer = transformerFactory.newTransformer(new StreamSource(new File(
7 "DOMParse09V4M0/09V4M0styling.xslt")));
8     transformer.setOutputProperty(OutputKeys.STANDALONE, "no");
9     transformer.setOutputProperty(OutputKeys.INDENT, "yes");
10    DOMSource source = new DOMSource(doc);
11    StreamResult result = new StreamResult(output);
12    transformer.transform(source, result);
13 }
14 }
15 }
```

## 2c) DomQuery:

Az xPath alkalmazásával kérek le adatokat az XML-ből.

A képeket az importok után kezdem.



```

1 public class DOMQuery09V4M0 {
2     private static XPath xPath = XPathFactory.newInstance().newXPath();
3     private static Document document;
4
5     public static void main(String[] args) {
6         try {
7             DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
8             DocumentBuilder dBuilder;
9             dBuilder = dbFactory.newDocumentBuilder();
10
11             document = dBuilder.parse("./XML09V4M0.xml");
12
13             document.getDocumentElement().normalize();
14
15             //Az osszes teljesitmeny igazolas lekerese.
16             queryTeljesitmenyIgazolas();
17
18             System.out.printf("-----\n\n");
19
20             //Megrendelesek szuere ID alapjan
21             queryMegrendelesekByID("RID02");
22
23             System.out.printf("-----\n\n");
24
25             //3-as szamu orvos elvezett rendeleseinek lekerese
26             queryHataridoNaploByDoctor(3);
27
28             System.out.printf("-----\n\n");
29
30             //18000Ft folotti bevetelek lekerdezese
31             queryBevetelByOsszeg(18000);
32
33             } catch (SAXException e) {
34                 e.printStackTrace();
35             } catch (ParserConfigurationException e) {
36                 e.printStackTrace();
37             } catch (IOException e) {
38                 e.printStackTrace();
39             }
40         }
41     }

```

```

1 private static void queryTeljesitmenyIgazolas(){
2     String expression = String.format("/egeszsegugy/teljesitmeny_igazolas");
3     try {
4         System.out.println("Osszes Teljesitmeny_Igazolas lekerdezese:");
5         NodeList nodeList = (NodeList) xPath.compile(expression).evaluate(document, XPathConstants.NODESET);
6
7         for (int i = 0; i < nodeList.getLength(); i++) {
8
9             Node nNode = nodeList.item(i);
10
11             if (nNode.getNodeType() == Node.ELEMENT_NODE) {
12
13                 Element element = (Element) nNode;
14                 String tid = element.getAttribute("TID");
15                 System.out.println("Azonosito: " + tid +
16                                     "\n\t Orvos azonosito: " + element.getElementsByTagName("orvos").item(0).getTextContent() +
17                                     "\n\t Fizetes: " + element.getElementsByTagName("fizetes").item(0).getTextContent() +
18                                     "\n\t Datum: " + element.getElementsByTagName("datum").item(0).getTextContent());
19             }
20         }
21     } catch (XPathExpressionException e) {
22         e.printStackTrace();
23     }
24 }
25
26 }
27

```

```

1 private static void queryMegrendelesekByID(String ID) {
2     String expression = String.format("/egeszssegugy/megrendelesek[RID = '%s']", ID);
3     try {
4         System.out.printf("Id alapjan lekerdezes: \n");
5
6         NodeList nodeList = (NodeList) XPath.compile(expression).evaluate(document, XPathConstants.NODESET);
7
8         for (int i = 0; i < nodeList.getLength(); i++) {
9
10            Node nNode = nodeList.item(i);
11
12            if (nNode.getNodeType() == Node.ELEMENT_NODE) {
13
14                Element element = (Element) nNode;
15                String rid = element.getAttribute("RID");
16                System.out.println("Azonosito: " + rid +
17                    "\n\t Megnevezes: " + element.getElementsByTagName("megnevezes").item(0).getTextContent() +
18                    "\n\t Datum: " + element.getElementsByTagName("datum").item(0).getTextContent() +
19                    "\n\t Penzkiadas: " + element.getElementsByTagName("penzkiadas").item(0).getTextContent() +
20                    "\n\t Darab: " + element.getElementsByTagName("darab").item(0).getTextContent());
21            }
22        }
23    } catch (XPathExpressionException e) {
24        e.printStackTrace();
25    }
26 }
27
28

```

```

1 private static void queryBevetelByOsszeg(int minimumPrice) {
2     String expression = String.format("/egeszssegugy/bevetel[osszeg>%s]", minimumPrice);
3     try {
4         System.out.printf("%s -tol nagyobb bevetelu szakrendelesek: \n");
5
6         NodeList nodeList = (NodeList) XPath.compile(expression).evaluate(document, XPathConstants.NODESET);
7
8         for (int i = 0; i < nodeList.getLength(); i++) {
9
10            Node nNode = nodeList.item(i);
11
12            if (nNode.getNodeType() == Node.ELEMENT_NODE) {
13                Element element = (Element) nNode;
14                String bid = element.getAttribute("BID");
15                String hid = element.getAttribute("HID");
16                System.out.println(
17                    "Azonositok: " + bid + " " + hid +
18                    "\n\t Megnevezes: " + element.getElementsByTagName("megnevezes").item(0).getTextContent() +
19                    "\n\t Osszeg: " + element.getElementsByTagName("osszeg").item(0).getTextContent());
20            }
21        }
22    } catch (XPathExpressionException e) {
23        e.printStackTrace();
24    }
25 }
26
27

```

```

1 private static void queryHataridoNaploByDoctor(int ID){
2     String expression = String.format("/egeszssegugy/hatarido_naplo");
3     try {
4         System.out.printf("3-as szamu orvos elvezgett rendelesei: \n");
5
6         NodeList nodeList = (NodeList) XPath.compile(expression).evaluate(document, XPathConstants.NODESET);
7
8         for (int i = 0; i < nodeList.getLength(); i++) {
9
10            Node nNode = nodeList.item(i);
11
12            if (nNode.getNodeType() == Node.ELEMENT_NODE) {
13                Element element = (Element) nNode;
14                if(Integer.parseInt(element.getElementsByTagName("szakorv").item(0).getTextContent()) == ID) {
15                    Node paciens = (Node) element.getElementsByTagName("paciens");
16                    System.out.println(
17                        "Idopont: "+element.getElementsByTagName("ido pont").item(0).getTextContent()+
18                        "Bevetel: "+element.getElementsByTagName("bevetel").item(0).getTextContent()+
19                        "Szakrendelo: "+element.getElementsByTagName("szakrend").item(0).getTextContent()+
20                        "Paciens neve: "+paciens.getAttributes().getNamedItem("nev").getTextContent()+
21                        "Paciens TAJszama: "+paciens.getAttributes().getNamedItem("TAJszam").getTextContent()+
22                        "Paciens telefonszama: "+paciens.getAttributes().getNamedItem("telefonszam").getTextContent()
23                    );
24                }
25            }
26        }
27    } catch (XPathExpressionException e) {
28        e.printStackTrace();
29    }
30 }
31
32 }
33

```