

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Méhészet

Készítette: **Boján Miron Noel**

Neptunkód: **F4XQRO**

Dátum: **2024.11.20**

Tartalomjegyzék

1. [Bevezetés](#)
- 1.1 [Az adatbázis ER modell tervezése](#)
- 1.2 [Az adatbázis konvertálása XDM modellre](#)
- 1.3 [Az XDM modell alapján XML dokumentum készítése](#)
- 1.4 [Az XML dokumentum alapján XMLSchema készítése](#)
2. [A második feladat](#)
- 2.1 [Adatolvasás](#)
- 2.2 [Adatírás](#)
- 2.3 [Adatkérdezés](#)
- 2.4 [Adatmódosítás](#)

1.Bevezetés

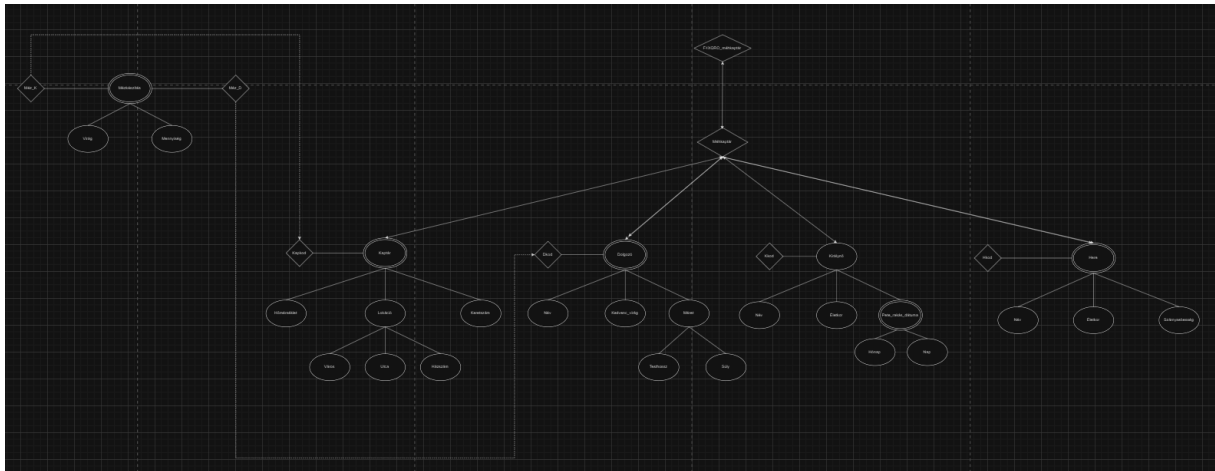
A feladat leírása

A fél éves feladatom célja egy méhészet működésének és felépítésének bemutatása adatmodell segítségével. A projekt során egy olyan rendszert dolgozok ki, amely a méhek életének, a kaptár működésének, valamint a méhészeti folyamatok adatainak rendszerezett és áttekinthető leírását biztosítja XML alapú formátumban.

A feladat során figyelembe veszem a méhészet fontos elemeit, például a különböző méhtípusokat (királynő, dolgozó, here), a kaptár felépítését és a méztermeléshez kapcsolódó adatokat.

1.1 Az adatbázis ER modell tervezése

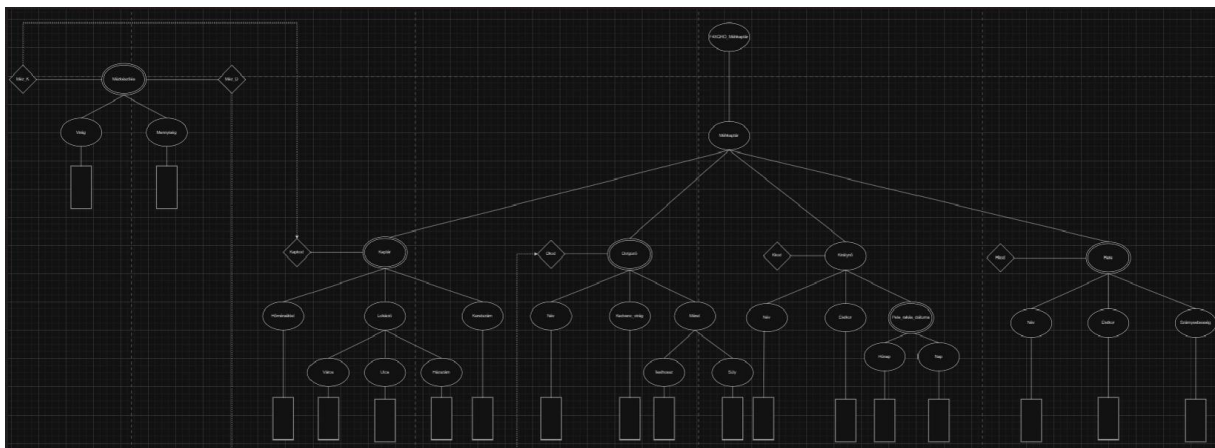
először egy ER modellt készítettem, amely a rendszer alapvető elemeit és azok közötti kapcsolatokat írja le.



(1.kép:ER modell)

1.2 Az adatbázis konvertálása XDM modellre

a relációs adatbázis struktúráját egy hierarchikus XDM modellbe konvertáltam.



(2.kép: XDM modell)

1.3 Az XDM modell alapján XML dokumentum készítése

A modell egyszerű és jól strukturált. Az XML dokumentumban a kaptárak, méhek (királynők, dolgozók, herék), valamint a mézkészítés folyamata szerepel. A kaptárak tartalmazzák a hőmérsékletet és a helyszín adatokat, a méhek egyedi jellemzőket, például méretekkel, életkorral és kedvenc virágokkal rendelkeznek.

```

1. <?xml version="1.0" encoding="UTF-8"?>
2.
3. <meheszet>
4.   <!--Kaptárak-->
5.     <kaptar kapkod = "kap1">
6.       <homerseklet>32</homerseklet>
7.       <lokacio>

```

```
8.         <varos>Győr</varos>
9.         <utca>Szondi</utca>
10.        <hazszam>5</hazszam>
11.        </lokacio>
12.        <keretszam>16</keretszam>
13.    </kaptar>
14.
15.    <kaptar kapkod = "kap2">
16.        <homerseklet>32</homerseklet>
17.        <lokacio>
18.            <varos>Sopron</varos>
19.            <utca>Kossuth</utca>
20.            <hazszam>15</hazszam>
21.        </lokacio>
22.        <keretszam>8</keretszam>
23.    </kaptar>
24.
25.    <kaptar kapkod = "kap3">
26.        <homerseklet>36</homerseklet>
27.        <lokacio>
28.            <varos>Szeged</varos>
29.            <utca>Petőfi</utca>
30.            <hazszam>46</hazszam>
31.        </lokacio>
32.        <keretszam>32</keretszam>
33.    </kaptar>
34.
35.    <!-- Királynők-->
36.    <kiralyno Kkod = "ki1">
37.        <nev>ZümZüm</nev>
38.        <eletkor>1</eletkor>
39.        <peterakasdatum>
40.            <honap>Május</honap>
41.            <nap>1</nap>
42.        </peterakasdatum>
43.    </kiralyno>
44.
45.    <kiralyno Kkod = "ki2">
46.        <nev>Arany szárny</nev>
47.        <eletkor>3</eletkor>
48.        <peterakasdatum>
49.            <honap>Június</honap>
50.            <nap>3</nap>
51.        </peterakasdatum>
52.    </kiralyno>
53.
54.    <kiralyno Kkod = "ki3">
55.        <nev>Bíborka</nev>
56.        <eletkor>3</eletkor>
57.        <peterakasdatum>
58.            <honap>Április</honap>
59.            <nap>23</nap>
60.        </peterakasdatum>
61.    </kiralyno>
62.
63.    <!-- Dolgozók-->
64.    <dolgozo Dkod = "do1">
65.        <nev>Virágszirom Vilmos</nev>
66.        <kedvenccirag>Napraforgó</kedvenccirag>
```

```

67.         <meret>
68.             <testhossz>4</testhossz>
69.             <suly>6</suly>
70.         </meret>
71.     </dolgozo>
72.
73.     <dolgozo Dkod = "do2">
74.         <nev>Sejtépítő Sándor</nev>
75.         <kedvenccirag>Akác</kedvenccirag>
76.         <meret>
77.             <testhossz>5</testhossz>
78.             <suly>8</suly>
79.         </meret>
80.     </dolgozo>
81.
82.     <dolgozo Dkod = "do3">
83.         <nev>Szorgos Szilárd</nev>
84.         <kedvenccirag>Repce</kedvenccirag>
85.         <meret>
86.             <testhossz>7</testhossz>
87.             <suly>4</suly>
88.         </meret>
89.     </dolgozo>
90.
91.     <!-- Herék-->
92.     <here Hkod="he1">
93.         <nev>Pollen Péter</nev>
94.         <eletkor>2</eletkor>
95.         <szarnysebesseg>11000/s</szarnysebesseg>
96.     </here>
97.
98.     <here Hkod="he2">
99.         <nev>Rovar Róbert</nev>
100.        <eletkor>1</eletkor>
101.        <szarnysebesseg>11400/s</szarnysebesseg>
102.    </here>
103.
104.    <here Hkod="he2">
105.        <nev>Gyűjtögető György</nev>
106.        <eletkor>3</eletkor>
107.        <szarnysebesseg>10000/s</szarnysebesseg>
108.    </here>
109.
110.    <!-- Mézkészítés-->
111.    <mezkeszites>
112.        <virag>Akác</virag>
113.        <Mennyiség>1</Mennyiség>
114.    </mezkeszites>
115.
116.    <mezkeszites>
117.        <virag>Napraforgó</virag>
118.        <Mennyiség>5</Mennyiség>
119.    </mezkeszites>
120.
121.    <mezkeszites>
122.        <virag>Levendula</virag>
123.        <Mennyiség>3</Mennyiség>
124.    </mezkeszites>
125. </mehesztes>

```

1.4 Az XML dokumentum alapján XMLSchema készítése

Az XML Schema a méhészeti adatainak struktúráját és érvényességi szabályait határozza meg.

```

1. <?xml version="1.0" encoding="UTF-8"?>
2.
3. <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
4.
5.     <!-- Egyszerű típusok -->
6.     <xs:simpleType name="pozitivEgesz">
7.         <xs:restriction base="xs:positiveInteger"/>
8.     </xs:simpleType>
9.
10.    <xs:simpleType name="honapTipus">
11.        <xs:restriction base="xs:string">
12.            <xs:enumeration value="Január"/>
13.            <xs:enumeration value="Február"/>
14.            <xs:enumeration value="Március"/>
15.            <xs:enumeration value="Április"/>
16.            <xs:enumeration value="Május"/>
17.            <xs:enumeration value="Június"/>
18.            <xs:enumeration value="Július"/>
19.            <xs:enumeration value="Augusztus"/>
20.            <xs:enumeration value="Szeptember"/>
21.            <xs:enumeration value="Október"/>
22.            <xs:enumeration value="November"/>
23.            <xs:enumeration value="December"/>
24.        </xs:restriction>
25.    </xs:simpleType>
26.
27.    <!-- Összetett típusok -->
28.    <xs:complexType name="lokacioTipus">
29.        <xs:sequence>
30.            <xs:element name="varos" type="xs:string"/>
31.            <xs:element name="utca" type="xs:string"/>
32.            <xs:element name="hazszam" type="pozitivEgesz"/>
33.        </xs:sequence>
34.    </xs:complexType>

```

```

35.
36.     <xs:complexType name="meretTipus">
37.         <xs:sequence>
38.             <xs:element name="testhossz" type="pozitivEgesz"/>
39.             <xs:element name="suly" type="pozitivEgesz"/>
40.         </xs:sequence>
41.     </xs:complexType>
42.
43.     <xs:complexType name="peterakasDatumTipus">
44.         <xs:sequence>
45.             <xs:element name="honap" type="honapTipus"/>
46.             <xs:element name="nap" type="pozitivEgesz"/>
47.         </xs:sequence>
48.     </xs:complexType>
49.
50.
51.     <xs:element name="meheszet">
52.         <xs:complexType>
53.             <xs:sequence>
54.                 <!-- Kaptárak -->
55.                 <xs:element name="kaptar" maxOccurs="unbounded">
56.                     <xs:complexType>
57.                         <xs:sequence>
58.                             <xs:element name="homerseklet"
type="pozitivEgesz"/>
59.                             <xs:element name="lokacio" type="lokacioTipus"/>
60.                             <xs:element name="keretszam"
type="pozitivEgesz"/>
61.                         </xs:sequence>
62.                         <xs:attribute name="kapkod" type="xs:ID"
use="required"/>
63.                     </xs:complexType>
64.                 </xs:element>
65.
66.                 <!-- Királynők -->
67.                 <xs:element name="kiralyno" maxOccurs="unbounded">
68.                     <xs:complexType>
69.                         <xs:sequence>
70.                             <xs:element name="nev" type="xs:string"/>
71.                             <xs:element name="eletkor" type="pozitivEgesz"/>
72.                             <xs:element name="peterakasdatum"
type="peterakasDatumTipus"/>
73.                         </xs:sequence>
74.                         <xs:attribute name="Kkod" type="xs:ID"
use="required"/>
75.                     </xs:complexType>
76.                 </xs:element>
77.
78.                 <!-- Dolgozók -->
79.                 <xs:element name="dolgozo" maxOccurs="unbounded">
80.                     <xs:complexType>
81.                         <xs:sequence>
82.                             <xs:element name="nev" type="xs:string"/>
83.                             <xs:element name="kedvenccirag"
type="xs:string"/>
84.                             <xs:element name="meret" type="meretTipus"/>
85.                         </xs:sequence>
86.                         <xs:attribute name="Dkod" type="xs:ID"
use="required"/>

```



```

87.         </xs:complexType>
88.     </xs:element>
89.
90.     <!-- Herék -->
91.     <xs:element name="here" maxOccurs="unbounded">
92.         <xs:complexType>
93.             <xs:sequence>
94.                 <xs:element name="nev" type="xs:string"/>
95.                 <xs:element name="eletkor" type="pozitivEgesz"/>
96.                 <xs:element name="szarnysebesseg"
type="xs:string"/>
97.             </xs:sequence>
98.             <xs:attribute name="Hkod" type="xs:ID"
use="required"/>
99.         </xs:complexType>
100.     </xs:element>
101.
102.     <!-- Mézkeszítés -->
103.     <xs:element name="mezkeszites" maxOccurs="unbounded">
104.         <xs:complexType>
105.             <xs:sequence>
106.                 <xs:element name="virag" type="xs:string"/>
107.                 <xs:element name="mennyiseg"
type="pozitivEgesz"/>
108.             </xs:sequence>
109.         </xs:complexType>
110.     </xs:element>
111. </xs:sequence>
112. </xs:complexType>
113. </xs:element>
114.
115. </xs:schema>
116.

```

2. A második feladat

2.1 adatolvasás

A program jelenleg lépésről lépésre olvassa be az XML fájlban található "here" elemek adatait.

```

1. for (int i = 0; i < nodeList.getLength(); i++) {
2.     Node node = nodeList.item(i);
3.     if (node.getNodeType() == Node.ELEMENT_NODE) {
4.         Element elem = (Element) node;
5.         String neve = elem.getElementsByTagName("nev").item(0).getTextContent();
6.         String kor = elem.getElementsByTagName("eletkor").item(0).getTextContent();
7.         String szarnyseb =
elem.getElementsByTagName("szarnysebesseg").item(0).getTextContent();
8.

```

Az `elem.getElementsByTagName` segítségével kiolvashatók az "nev", "eletkor" és "szarnysebesség" mezők értékei.

<https://github.com/mironprog/XML-SemTask/blob/main/2.feladat/DOMParseF4XQRO/src/DOMReadF4XQRO.java>

2.2 Adatírás

Ez a program jól mutatja be a DOM API használatát az XML dokumentumok feldolgozására. A Transformer osztály segítségével könnyen szerkeszthető az XML fájl.

```
1. TransformerFactory transformerFactory = TransformerFactory.newInstance();
2.         Transformer transformer = transformerFactory.newTransformer();
3.         transformer.setOutputProperty(OutputKeys.INDENT, "yes");
4.
```

A DOM API Transformer osztályát arra használjuk, hogy egy XML dokumentum DOM objektumát szöveges formátumú XML dokumentummá alakítsuk.

<https://github.com/mironprog/XML-SemTask/blob/main/2.feladat/DOMParseF4XQRO/src/DOMWriteF4XQRO.java>

2.3 Adatkérdezés

A program DOM API-t használ az XML dokumentum elemzéséhez, és négy különböző lekérdezést hajt végre, amelyek során az elemek adatait gyűjti ki.

Az eredményeket strukturáltan jeleníti meg a konzolon, lehetővé téve az XML dokumentumban tárolt információk olvashatóságát.

```
1. System.out.println("\n1. lekérdezés: Kaptárak és azonosítók:");
2.     NodeList kaptarList = doc.getElementsByTagName("kaptar");
3.     for (int i = 0; i < kaptarList.getLength(); i++) {
4.         Node kaptarNode = kaptarList.item(i);
5.         if (kaptarNode.getNodeType() == Node.ELEMENT_NODE) {
6.             Element kaptarElement = (Element) kaptarNode;
7.             System.out.println("Kaptár azonosító: " +
kaptarElement.getAttribute("kapkod"));
8.         }
}
```

```
9.         }  
10.
```

Ez a rész az XML dokumentumból az összes "kaptar" elem tagjait gyűjti össze és kiírja az azokhoz tartozó ID értékeket, amelyek az egyes kaptárak azonosítóit tartalmazzák.

<https://github.com/mironprog/XML-SemTask/blob/main/2.feladat/DOMParseF4XQRO/src/DOMQueryF4XQRO.java>

2.4 Adatmódosítás

A program betölt egy XML dokumentumot és különféle módosításokat hajt végre. A program egy új "dolgozo" elem tulajdonságát is frissíti, például kedvenc virágának nevét megváltoztatja, majd az eredményül kapott dokumentumot egy új fájlba menti a Transformer API segítségével.

```
1. NodeList kaptarList = doc.getElementsByTagName("kaptar");  
2.         if (kaptarList.getLength() > 0) {  
3.             Element firstKaptar = (Element) kaptarList.item(0);  
4.             firstKaptar.getElementsByTagName("homerseklet").item(0).setText  
Content("35");  
5.             System.out.println("1. Módosítás: Az első kaptár hőmérséklete  
35-re módosítva.");  
6.         }  
7.
```

A kód először lekéri az összes "kaptar" elemet az XML dokumentumból, majd ellenőrzi, hogy van-e ilyen elem. Ha található, a legelső "kaptar" elem "homerseklet" értékét 35-re módosítja.

<https://github.com/mironprog/XML-SemTask/blob/main/2.feladat/DOMParseF4XQRO/src/DOMModifyF4XQRO.java>