Jegyzőkönyv

Adatkezelés XML környezetben

Féléves feladat

Horgász Verseny

Készítette: **Koubridis Michael**

Neptunkód: **LWUIJ2**

Dátum: 2024.12.09

Tartalom

[Feladat leírása: 3](#_Toc184730830)

[1 Az adatbázis ER modell 4](#_Toc184730831)

[1.1 Egyedek és az ER modell 4](#_Toc184730832)

[1.2 Adatbázis ER modell 6](#_Toc184730833)

[1.3 Az XDM modell alapján XML dokumentum készítése 7](#_Toc184730834)

[1.4 Az XML dokumentum alapján XMLSchema készítése 10](#_Toc184730835)

[2 DOM program 14](#_Toc184730836)

[2.1 DOM adatolvasás 14](#_Toc184730837)

[2.2 DOM Adat módosítás 17](#_Toc184730838)

[2.3 Adatlekérdezés 19](#_Toc184730839)

[2.4 DomWrite 20](#_Toc184730840)

# Feladat leírása:

Feladatom egy olyan adatbázis ER modelljének a megtervezése, ami megfelel a normálformáknak és nem tartalmaz redundanciát. Az adatbázis megfelelő lesz a horgászversenyen történt események tárolására. Eredmények lekérdezésére akár az adottkifogott hal fajtának kifogott mennyiségére vagy versenyzőre lebontva. Ki mennyit, mit és mikor fogta a halat. Persze látható az adatbázisban, hogy kik vettek részt a versenyen és kik voltak a versenybírók. A hal típus jelöli, hogy milyen típusú a hal például keszeg. Vannak neki tulajdonságai típusa és az indexe ezenfelül, hogy őshonos-e és persze ragadozó vagy növényevő-e. Az index elsődleges kulcs. A kifogott hal egyed tartalmazza a kifogott halat ki mikor és mekkorát fogott. A Versenyző egyed tartalmazza a versenyző adatait, amihez kapcsolódik a díj egyed, ami tartalmazza azokat a tulajdonságokat, hogy melyik versenyző milyen díjakat nyert el. A helyszín egyed, ami kapcsolódik a versenyzőhöz tartalmazza, hogy melyik versenyző melyik horgász helyen ült. A verseny bíró egyed tartalmazza, hogy melyik versenyzőnél volt jelen az adott bíró, és a bírónak is el van tárolva a neve meg a születési adatok.

# 1 Az adatbázis ER modell

Horgászverseny ER modell tervezését és készítéséhez egy online EE modell készítő felület használtam, aminek a neve ERDPlus. Az ERDPlus-ba könnyen lehet létrehozni egyedekett, és az azokhoz tartalmazó attribútumokat. Ezenfelül könnyen lehet benne ábrázolni az egyedek között lévő kapcsolatokat.

## 1.1 Egyedek és az ER modell

**Hal típus:**

* **Index:** A hal típus egyed elsődleges kulcsa.
* **Típusa:** A hal védett hal-e.
* **Őshonos:** Az adott hal-e őshonos.
* **Ragadozó:** A hal ragadozó-e

**Kifogott hal:**

* **Hazon:** A kifogott hal elsődleges kulcsa.
* **Kg:** A hal súlya.
* **THal:** A hal fajtája

**Versenyző:**

* **VAzon:** A versenyző egyed elsődleges kulcsa.
* **VNév:** A versenyző neve.
* **Sz\_adatok:**
  + **Szul\_ido:** A versenyző születési ideje.
  + **Szul\_hely:** A versenyző születési helye.
* **Doksi:** A versenyző személyes iratai

**Helyszín:**

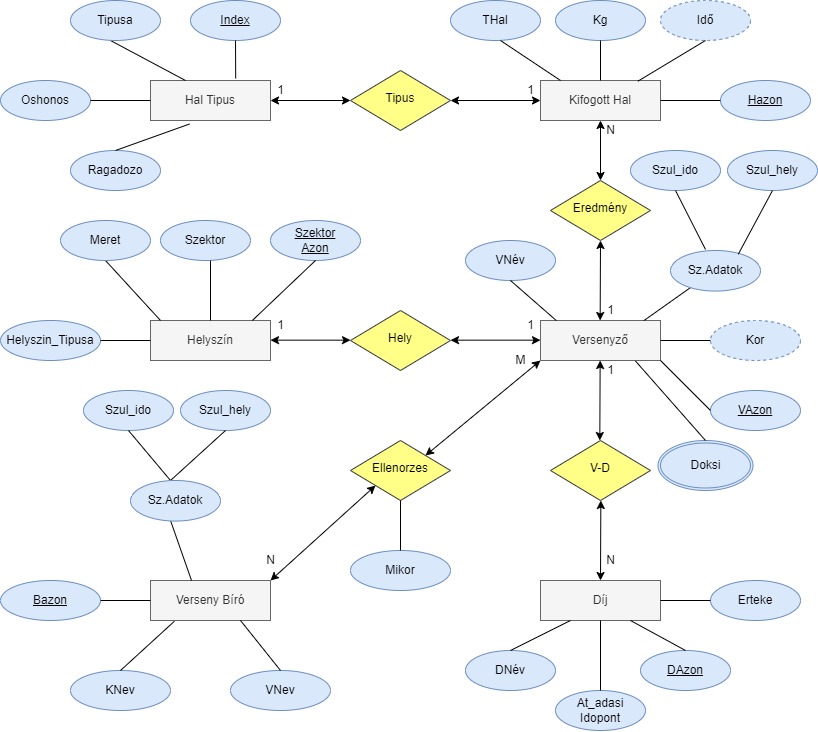
* **Szektor azon:** A helyszín egyed elsődleges kulcsa.
* **Szektor:** A szektor neve
* **Méret:** A szektor mérete.
* **Helyszín típusa:** Magas vagy alacsony part-e.

**Verseny Bíró:**

* **Bazon:** A verseny bíró egyed elsődleges kulcsa
* **VNev:** A bíró vezeték neve.
* **Knev:** A bíró kereszt neve.
* **Sz\_Adatok**
  + **Szul\_ido:** A biró születési ideje
  + **Szul\_hely:** A biró születési helye

**Díj:**

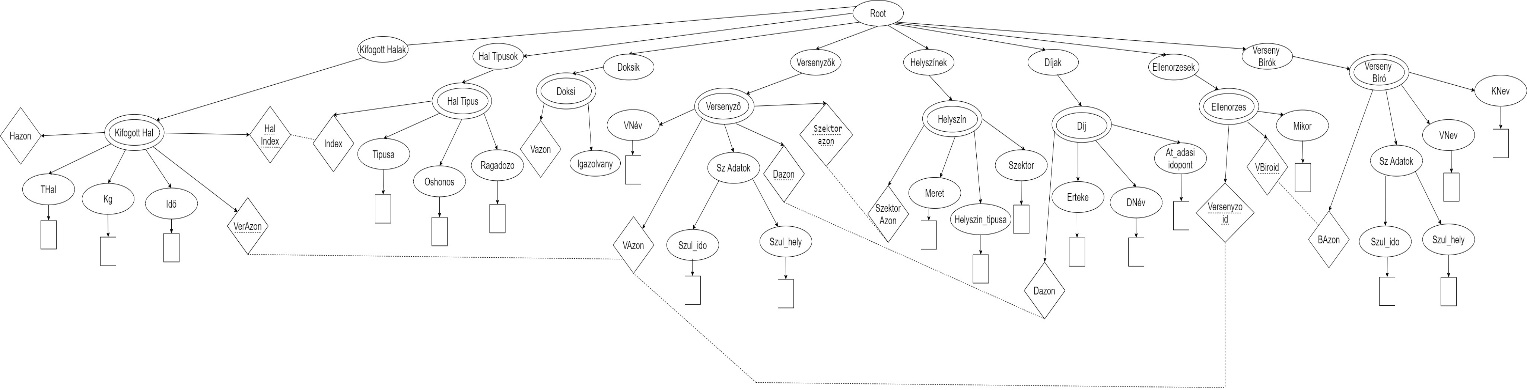
* **DAzon:** A díj egyed elsődleges kulcsa
* **DNév**: A díj neve
* **At\_adasi idopont**: Mikor lett átadva a díj
* **Erteke**: A díjnak az értéke



1. ábra: A horgászverseny ER modellje

## 1.2 Adatbázis ER modell

ER modellt XDM modellre úgy konvertálhatunk, hogy három féle jelölést használunk. Van az elem, amely az XDM modellben ellipszissel lehet ábrázolni. Az attribútumot rombusszal lehet jelölni, illetve a szöveget a téglalapban. Ezeken felül van még a dupla ellipszis, amely a többszörös előfordulási lehetőséget határozza meg. Ezenkívül minden egyedhez és N:M kapcsolathoz felvesszük az egyed többesszám alakját is.



2. ábra: Horgászverseny xdm modellje

## 1.3 Az XDM modell alapján XML dokumentum készítése

*<?***xml version="1.0" encoding="UTF-8"***?>*<**root xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="XMLSchemaLWUIJ2.xsd"**>  
 <**Versenyzok**>  
 <**Versenyzo VAzon="1" Szektorazon="1" Dazon="1"**>  
 <**VNev**>Bence</**VNev**>  
 <**Sz\_adatok**>  
 <**Szul\_ido**>1997.01.06</**Szul\_ido**>  
 <**Szul\_hely**>Gyula</**Szul\_hely**>  
 </**Sz\_adatok**>  
 </**Versenyzo**>  
 <**Versenyzo VAzon="2" Szektorazon="2" Dazon="2"**>  
 <**VNev**>Akos</**VNev**>  
 <**Sz\_adatok**>  
 <**Szul\_ido**>2000.10.15</**Szul\_ido**>  
 <**Szul\_hely**>Miskolc</**Szul\_hely**>  
 </**Sz\_adatok**>  
 </**Versenyzo**>  
 <**Versenyzo VAzon="3" Szektorazon="3" Dazon="3"**>  
 <**VNev**>Alex</**VNev**>  
 <**Sz\_adatok**>  
 <**Szul\_ido**>1999.03.20</**Szul\_ido**>  
 <**Szul\_hely**>Budapest</**Szul\_hely**>  
 </**Sz\_adatok**>  
 </**Versenyzo**>  
 </**Versenyzok**>  
 <**Doksik**>  
 <**Doksi Vazon="1"**>  
 <**Igazolvany**>Taj\_Kartya</**Igazolvany**>  
 </**Doksi**>  
 <**Doksi Vazon="2"**>  
 <**Igazolvany**>Vezetoi\_Engedely</**Igazolvany**>  
 </**Doksi**>  
 <**Doksi Vazon="3"**>  
 <**Igazolvany**>Szig\_szam</**Igazolvany**>  
 </**Doksi**>  
 <**Doksi Vazon="4"**>  
 <**Igazolvany**>Lakcim\_kartya</**Igazolvany**>  
 </**Doksi**>  
 </**Doksik**>  
 <**Verseny\_Birok**>  
 <**Verseny\_Biro BAzon="1"**>  
 <**VNev**>Kiss</**VNev**>  
 <**KNev**>Joe</**KNev**>  
 <**Sz\_adatok**>  
 <**Szul\_ido**>1980.06.20</**Szul\_ido**>  
 <**Szul\_hely**>Budapest</**Szul\_hely**>  
 </**Sz\_adatok**>  
 </**Verseny\_Biro**>  
 <**Verseny\_Biro BAzon="2"**>  
 <**VNev**>Nagy</**VNev**>  
 <**KNev**>Ferenc</**KNev**>  
 <**Sz\_adatok**>  
 <**Szul\_ido**>1985.12.03</**Szul\_ido**>  
 <**Szul\_hely**>Sopron</**Szul\_hely**>  
 </**Sz\_adatok**>  
 </**Verseny\_Biro**>  
 <**Verseny\_Biro BAzon="3"**>  
 <**VNev**>Kovacs</**VNev**>  
 <**KNev**>Janos</**KNev**>  
 <**Sz\_adatok**>  
 <**Szul\_ido**>1975.02.10</**Szul\_ido**>  
 <**Szul\_hely**>Gyor</**Szul\_hely**>  
 </**Sz\_adatok**>  
 </**Verseny\_Biro**>  
 </**Verseny\_Birok**>  
 <**Kifogott\_Halak**>  
 <**Kifogott\_hal Hazon="1" Halindex="1" VerAzon="1"**>  
 <**THal**>Csuka</**THal**>  
 <**Kg**>5</**Kg**>  
 <**ido**>14:00</**ido**>  
 </**Kifogott\_hal**>  
 <**Kifogott\_hal Hazon="2" Halindex="2" VerAzon="2"**>  
 <**THal**>Ponty</**THal**>  
 <**Kg**>10</**Kg**>  
 <**ido**>15:00</**ido**>  
 </**Kifogott\_hal**>  
 <**Kifogott\_hal Hazon="3" Halindex="3" VerAzon="3"**>  
 <**THal**>Amur</**THal**>  
 <**Kg**>8</**Kg**>  
 <**ido**>17:00</**ido**>  
 </**Kifogott\_hal**>  
 </**Kifogott\_Halak**>  
 <**Ellenorzesek**>  
 <**Ellenorzes VersenyzoId="2" VBiroId="1"**>  
 <**Mikor**>17:55</**Mikor**>  
 </**Ellenorzes**>  
 <**Ellenorzes VersenyzoId="1" VBiroId="1"**>  
 <**Mikor**>17:50</**Mikor**>  
 </**Ellenorzes**>  
 <**Ellenorzes VersenyzoId="3" VBiroId="3"**>  
 <**Mikor**>17:45</**Mikor**>  
 </**Ellenorzes**>  
 <**Ellenorzes VersenyzoId="2" VBiroId="3"**>  
 <**Mikor**>17:40</**Mikor**>  
 </**Ellenorzes**>  
 <**Ellenorzes VersenyzoId="1" VBiroId="2"**>  
 <**Mikor**>17:35</**Mikor**>  
 </**Ellenorzes**>  
 </**Ellenorzesek**>  
 <**Dijak**>  
 <**Dij Dazon="1"**>  
 <**DNev**>Legtobb\_Hal</**DNev**>  
 <**Erteke**>5000</**Erteke**>  
 <**At\_adasi\_idopont**>18:00</**At\_adasi\_idopont**>  
 </**Dij**>  
 <**Dij Dazon="2"**>  
 <**DNev**>Legnagyobb\_Hal</**DNev**>  
 <**Erteke**>5000</**Erteke**>  
 <**At\_adasi\_idopont**>18:30</**At\_adasi\_idopont**>  
 </**Dij**>  
 <**Dij Dazon="3"**>  
 <**DNev**>Gyoztes</**DNev**>  
 <**Erteke**>20000</**Erteke**>  
 <**At\_adasi\_idopont**>19:00</**At\_adasi\_idopont**>  
 </**Dij**>  
 </**Dijak**>  
 <**Helyszinek**>  
 <**Helyszin SzektorAzon="1"**>  
 <**Helyszin\_tipusa**>Alacsony\_part</**Helyszin\_tipusa**>  
 <**Meret**>2033</**Meret**>  
 <**Szektor**>A</**Szektor**>  
 </**Helyszin**>  
 <**Helyszin SzektorAzon="2"**>  
 <**Helyszin\_tipusa**>Magas\_part</**Helyszin\_tipusa**>  
 <**Meret**>25</**Meret**>  
 <**Szektor**>B</**Szektor**>  
 </**Helyszin**>  
 <**Helyszin SzektorAzon="3"**>  
 <**Helyszin\_tipusa**>Alacsony\_part</**Helyszin\_tipusa**>  
 <**Meret**>30</**Meret**>  
 <**Szektor**>C</**Szektor**>  
 </**Helyszin**>  
 </**Helyszinek**>  
 <**Hal\_Tipusok**>  
 <**Hal\_Tipus index="1"**>  
 <**Ragadozo**>igen</**Ragadozo**>  
 <**Tipusa**>VedettHal</**Tipusa**>  
 <**Oshonos**>nem</**Oshonos**>  
 </**Hal\_Tipus**>  
 <**Hal\_Tipus index="2"**>  
 <**Ragadozo**>nem</**Ragadozo**>  
 <**Tipusa**>VedettHal</**Tipusa**>  
 <**Oshonos**>igen</**Oshonos**>  
 </**Hal\_Tipus**>  
 <**Hal\_Tipus index="3"**>  
 <**Ragadozo**>igen</**Ragadozo**>  
 <**Tipusa**>NemVedettHal</**Tipusa**>  
 <**Oshonos**>igen</**Oshonos**>  
 </**Hal\_Tipus**>  
 </**Hal\_Tipusok**>  
</**root**>

## 1.4 Az XML dokumentum alapján XMLSchema készítése

*<?***xml version="1.0" encoding="UTF-8"***?>*<**xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"**>  
 <**xs:element name="root"**>  
 <**xs:complexType**>  
 <**xs:sequence**>  
 <**xs:element name="Versenyzok"**>  
 <**xs:complexType**>  
 <**xs:sequence**>  
 <**xs:element name="Versenyzo" maxOccurs="1000"**>  
 <**xs:complexType**>  
 <**xs:sequence**>  
 <**xs:element name="VNev" type="xs:string"** />  
 <**xs:element name="Sz\_adatok"**>  
 <**xs:complexType**>  
 <**xs:sequence**>  
 <**xs:element name="Szul\_ido" type="szulido"** />  
 <**xs:element name="Szul\_hely" type="xs:string"** />  
 </**xs:sequence**>  
 </**xs:complexType**>  
 </**xs:element**>  
 </**xs:sequence**>  
 <**xs:attribute name="VAzon" type="xs:string"** />  
 <**xs:attribute name="Szektorazon" type="xs:string"** />  
 <**xs:attribute name="Dazon" type="xs:string"** />  
 </**xs:complexType**>  
 </**xs:element**>  
 </**xs:sequence**>  
 </**xs:complexType**>  
 </**xs:element**>  
 <**xs:element name="Doksik"**>  
 <**xs:complexType**>  
 <**xs:sequence**>  
 <**xs:element name="Doksi" maxOccurs="1000"**>  
 <**xs:complexType**>  
 <**xs:sequence**>  
 <**xs:element name="Igazolvany" type="xs:string"** />  
 </**xs:sequence**>  
 <**xs:attribute name="Vazon" type="xs:string"** />  
 </**xs:complexType**>  
 </**xs:element**>  
 </**xs:sequence**>  
 </**xs:complexType**>  
 </**xs:element**>  
 <**xs:element name="Verseny\_Birok"**>  
 <**xs:complexType**>  
 <**xs:sequence**>  
 <**xs:element name="Verseny\_Biro" minOccurs="1" maxOccurs="1000"**>  
 <**xs:complexType**>  
 <**xs:sequence**>  
 <**xs:element name="VNev" type="xs:string"** />  
 <**xs:element name="KNev" type="xs:string"** />  
 <**xs:element name="Sz\_adatok"**>  
 <**xs:complexType**>  
 <**xs:sequence**>  
 <**xs:element name="Szul\_ido" type="szulido"** />  
 <**xs:element name="Szul\_hely" type="xs:string"** />  
 </**xs:sequence**>  
 </**xs:complexType**>  
 </**xs:element**>  
 </**xs:sequence**>  
 <**xs:attribute name="BAzon" type="xs:string"**/>  
 </**xs:complexType**>  
 </**xs:element**>  
 </**xs:sequence**>  
 </**xs:complexType**>  
 </**xs:element**>  
 <**xs:element name="Kifogott\_Halak"**>  
 <**xs:complexType**>  
 <**xs:sequence**>  
 <**xs:element name="Kifogott\_hal" minOccurs="1" maxOccurs="1000"**>  
 <**xs:complexType**>  
 <**xs:sequence**>  
 <**xs:element name="THal" type="xs:string"**/>  
 <**xs:element name="Kg" type="xs:string"**/>  
 <**xs:element name="ido" type="xs:string"**/>  
 </**xs:sequence**>  
 <**xs:attribute name="Hazon" type="xs:string"**/>  
 <**xs:attribute name="Halindex" type="xs:string"**/>  
 <**xs:attribute name="VerAzon" type="xs:string"**/>  
 </**xs:complexType**>  
 </**xs:element**>  
 </**xs:sequence**>  
 </**xs:complexType**>  
 </**xs:element**>  
 <**xs:element name="Ellenorzesek"**>  
 <**xs:complexType**>  
 <**xs:sequence**>  
 <**xs:element name="Ellenorzes" minOccurs="1" maxOccurs="1000"**>  
 <**xs:complexType**>  
 <**xs:sequence**>  
 <**xs:element name="Mikor" type="idoType"**/>  
 </**xs:sequence**>  
 <**xs:attribute name="VersenyzoId" type="xs:string"**/>  
 <**xs:attribute name="VBiroId" type="xs:string"**/>  
 </**xs:complexType**>  
 </**xs:element**>  
 </**xs:sequence**>  
 </**xs:complexType**>  
 </**xs:element**>  
 <**xs:element name="Dijak"**>  
 <**xs:complexType**>  
 <**xs:sequence**>  
 <**xs:element name="Dij" minOccurs="1" maxOccurs="1000"**>  
 <**xs:complexType**>  
 <**xs:sequence**>  
 <**xs:element name="DNev" type="xs:string"**/>  
 <**xs:element name="Erteke" type="xs:string"**/>  
 <**xs:element name="At\_adasi\_idopont" type="idoType"**/>  
 </**xs:sequence**>  
 <**xs:attribute name="Dazon" type="xs:string"**/>  
 </**xs:complexType**>  
 </**xs:element**>  
 </**xs:sequence**>  
 </**xs:complexType**>  
 </**xs:element**>  
 <**xs:element name="Helyszinek"**>  
 <**xs:complexType**>  
 <**xs:sequence**>  
 <**xs:element name="Helyszin" minOccurs="1" maxOccurs="1000"**>  
 <**xs:complexType**>  
 <**xs:sequence**>  
 <**xs:element name="Helyszin\_tipusa" type="xs:string"**/>  
 <**xs:element name="Meret" type="xs:positiveInteger"**/>  
 <**xs:element name="Szektor" type="xs:string"**/>  
 </**xs:sequence**>  
 <**xs:attribute name="SzektorAzon" type="xs:string"**/>  
 </**xs:complexType**>  
 </**xs:element**>  
 </**xs:sequence**>  
 </**xs:complexType**>  
 </**xs:element**>  
 <**xs:element name="Hal\_Tipusok"**>  
 <**xs:complexType**>  
 <**xs:sequence**>  
 <**xs:element name="Hal\_Tipus" minOccurs="1" maxOccurs="1000"**>  
 <**xs:complexType**>  
 <**xs:sequence**>  
 <**xs:element name="Ragadozo" type="xs:string"**/>  
 <**xs:element name="Tipusa" type="xs:string"**/>  
 <**xs:element name="Oshonos" type="xs:string"**/>  
 </**xs:sequence**>  
 <**xs:attribute name="index" type="xs:string"**/>  
 </**xs:complexType**>  
 </**xs:element**>  
 </**xs:sequence**>  
 </**xs:complexType**>  
 </**xs:element**>  
 </**xs:sequence**>  
 </**xs:complexType**>  
 <**xs:key name="dij\_Key"**>  
 <**xs:selector xpath="Dij"**/>  
 <**xs:field xpath="@Dazon"**/>  
 </**xs:key**>  
 <**xs:key name="Versenyzo\_Key"**>  
 <**xs:selector xpath="Versenyzo"**/>  
 <**xs:field xpath="@VAzon"**/>  
 </**xs:key**>  
 <**xs:key name="kifogottHal\_Key"**>  
 <**xs:selector xpath="Kifogott\_hal"**/>  
 <**xs:field xpath="@Hazon"**/>  
 </**xs:key**>  
 <**xs:key name="VersenyBir\_Key"**>  
 <**xs:selector xpath="Verseny\_Biro"**/>  
 <**xs:field xpath="@Bazon"**/>  
 </**xs:key**>  
 <**xs:key name="Hal\_Tipus\_key"**>  
 <**xs:selector xpath="Hal\_Tipus"**/>  
 <**xs:field xpath="@index"**/>  
 </**xs:key**>  
 <**xs:key name="helyszin\_key"**>  
 <**xs:selector xpath="Helyszin"**/>  
 <**xs:field xpath="@SzektorAzon"**/>  
 </**xs:key**>  
 <**xs:keyref name="Fkey\_dij\_Versenyzo" refer="dij\_Key"**>  
 <**xs:selector xpath="Versenyzo"**/>  
 <**xs:field xpath="@Dazon"**/>  
 </**xs:keyref**>  
 <**xs:keyref name="Fkey\_H-V\_VersenyBiro" refer="VersenyBir\_Key"**>  
 <**xs:selector xpath="Ellenorzes"**/>  
 <**xs:field xpath="@VBiroId"**/>  
 </**xs:keyref**>  
 <**xs:keyref name="Fkey\_H-V\_Versenyzo" refer="Versenyzo\_Key"**>  
 <**xs:selector xpath="Ellenorzes"**/>  
 <**xs:field xpath="@VersenyzoId"**/>  
 </**xs:keyref**>  
 <**xs:keyref name="Fkey\_Hal\_Tipus\_Kifogott\_hal" refer="Hal\_Tipus\_key"**>  
 <**xs:selector xpath="Kifogott\_hal"**/>  
 <**xs:field xpath="@Halindex"**/>  
 </**xs:keyref**>  
 <**xs:keyref name="Fkey\_Kifogott\_hal\_Versenyzo" refer="Versenyzo\_Key"**>  
 <**xs:selector xpath="Kifogott\_hal"**/>  
 <**xs:field xpath="@VerAzon"**/>  
 </**xs:keyref**>  
 <**xs:keyref name="Fkey\_Helyszin\_Versenyzo" refer="helyszin\_key"**>  
 <**xs:selector xpath="Versenyzo"**/>  
 <**xs:field xpath="@Szektorazon"**/>  
 </**xs:keyref**>  
 <**xs:keyref name="Fkey\_Doksi\_Versenyzo" refer="Versenyzo\_Key"**>  
 <**xs:selector xpath="Doksi"**/>  
 <**xs:field xpath="@Vazon"**/>  
 </**xs:keyref**>  
 </**xs:element**>  
 <**xs:simpleType name="idoType"**>  
 <**xs:restriction base="xs:string"**>  
 <**xs:pattern value="[0-9]{1}[0-9]{1}:[0-9]{1}[0-9]{1}"** />  
 </**xs:restriction**>  
 </**xs:simpleType**>  
 <**xs:simpleType name="szulido"**>  
 <**xs:restriction base="xs:string"**>  
 <**xs:pattern value="\d{4}[.]\d{2}[.]\{d2}"**/>  
 <**xs:length value="10"**/>  
 </**xs:restriction**>  
 </**xs:simpleType**>  
</**xs:schema**>

# 2 DOM program

A DOM programot Java környezetben készítettem el. A DOM program tartalmaz adat módosítást, adat lekérdezést, illetve adatolvasást is.

## 2.1 DOM adatolvasás

Dom segítségével ki listázza az összes adatott.

**package** hu.domparse.LWUIJ2;  
  
**import** java.io.File;  
**import** java.io.IOException;  
  
**import** javax.xml.parsers.DocumentBuilder;  
**import** javax.xml.parsers.DocumentBuilderFactory;  
**import** javax.xml.parsers.ParserConfigurationException;  
  
**import** org.w3c.dom.Document;  
**import** org.w3c.dom.Element;  
**import** org.w3c.dom.Node;  
**import** org.w3c.dom.NodeList;  
**import** org.xml.sax.SAXException;  
  
  
**public class** DomReadLWUIJ2 {  
 **public static void** main(String[] args){  
 NodeList list;  
 **try** {  
 DocumentBuilderFactory factory=DocumentBuilderFactory.*newInstance*();  
 DocumentBuilder documentBuilder=factory.newDocumentBuilder();  
 *//fájl beolvasása* Document document=documentBuilder.parse(**new** File(**"XMLLWUIJ2.xml"**));  
 document.getDocumentElement().normalize();  
 *//Gyökér elem megkeresése* System.***out***.println(**"Root element : "** + document.getDocumentElement().getNodeName());  
 System.***out***.println(**"----------------"**);  
 *//Aktuális elem meghatározása* list=document.getElementsByTagName(**"Versenyzo"**);  
  
 **for** (**int** i=0;i<list.getLength();i++) {  
 Node node=list.item(i);  
 System.***out***.println(**"\nAktuális elem: "** + node.getNodeName());  
 *//Versenyzok adatainak kiirása* **if**(node.getNodeType() == Node.***ELEMENT\_NODE***) {  
 Element element=(Element) node;  
 System.***out***.println(**"Versenyzo id: "** + element.getAttribute(**"VAzon"**));  
 System.***out***.println(**"Szektorazonositó id: "** + element.getAttribute(**"Szektorazon"**));  
 System.***out***.println(**"Díj id: "** + element.getAttribute(**"Dazon"**));  
 System.***out***.println(**"VersenyzőNév: "** + element.getElementsByTagName(**"VNev"**).item(0).getTextContent());  
  
 *//Versenyző születési adatainak kiirása* Node nodeszuladat=list.item(i);  
 **if**(nodeszuladat.getNodeType()==Node.***ELEMENT\_NODE***){  
 Element elementszuldat=(Element) node;  
 System.***out***.println(**"Születési idő: "** + elementszuldat.getElementsByTagName(**"Szul\_ido"**).item(0).getTextContent());  
 System.***out***.println(**"Születési hely: "** + elementszuldat.getElementsByTagName(**"Szul\_hely"**).item(0).getTextContent());  
 }  
 }  
 }  
 *//Aktuális elem meghatározása* list=document.getElementsByTagName(**"Doksi"**);  
  
 **for**(**int** i=0;i<list.getLength();i++){  
 Node node=list.item(i);  
 System.***out***.println(**"\nAktuális elem: "** + node.getNodeName());  
 *//Doksi adatainak kiirása* **if**(node.getNodeType() == Node.***ELEMENT\_NODE***) {  
 Element element=(Element) node;  
 System.***out***.println(**"Dokumentum id: "** + element.getAttribute(**"Vazon"**));  
 System.***out***.println(**"Igazolvany: "** + element.getElementsByTagName(**"Igazolvany"**).item(0).getTextContent());  
 }  
 }  
 *//Aktuális elem meghatározása* list=document.getElementsByTagName(**"Verseny\_Biro"**);  
  
 **for**(**int** i=0;i<list.getLength();i++){  
 Node node=list.item(i);  
 System.***out***.println(**"\nAktuális elem: "** + node.getNodeName());  
 *//Verseny Bírok adatainak kiirása* **if**(node.getNodeType() == Node.***ELEMENT\_NODE***) {  
 Element element=(Element) node;  
 System.***out***.println(**"Verseny Biró id: "** + element.getAttribute(**"BAzon"**));  
 System.***out***.println(**"Vezetéknév: "** + element.getElementsByTagName(**"VNev"**).item(0).getTextContent());  
 System.***out***.println(**"Keresztnév: "** + element.getElementsByTagName(**"KNev"**).item(0).getTextContent());  
  
 *//Biró születési adatainak kiirása* Node nodeszuladat=list.item(i);  
 **if**(nodeszuladat.getNodeType()==Node.***ELEMENT\_NODE***){  
 Element elementszuldat=(Element) node;  
 System.***out***.println(**"Születési idő: "** + elementszuldat.getElementsByTagName(**"Szul\_ido"**).item(0).getTextContent());  
 System.***out***.println(**"Születési hely: "** + elementszuldat.getElementsByTagName(**"Szul\_hely"**).item(0).getTextContent());  
 }  
 }  
 }  
 *//Aktuális elem meghatározása* list=document.getElementsByTagName(**"Kifogott\_hal"**);  
  
 **for**(**int** i=0;i<list.getLength();i++){  
 Node node=list.item(i);  
 System.***out***.println(**"\nAktuális elem: "** + node.getNodeName());  
 *//Doksi adatainak kiirása* **if**(node.getNodeType() == Node.***ELEMENT\_NODE***) {  
 Element element=(Element) node;  
 System.***out***.println(**"Hal id: "** + element.getAttribute(**"Vazon"**));  
 System.***out***.println(**"KifogottHalnak az idje: "** + element.getAttribute(**"Halindex"**));  
 System.***out***.println(**"Versenyző idje: "** + element.getAttribute(**"VerAzon"**));  
 System.***out***.println(**"Tipusa: "** + element.getElementsByTagName(**"THal"**).item(0).getTextContent());  
 System.***out***.println(**"Mérete: "** + element.getElementsByTagName(**"Kg"**).item(0).getTextContent());  
 System.***out***.println(**"Mikor fogták: "** + element.getElementsByTagName(**"ido"**).item(0).getTextContent());  
  
 }  
 }  
 *//Aktuális elem meghatározása* list=document.getElementsByTagName(**"Ellenorzes"**);  
  
 **for**(**int** i=0;i<list.getLength();i++){  
 Node node=list.item(i);  
 System.***out***.println(**"\nAktuális elem: "** + node.getNodeName());  
 *//Doksi adatainak kiirása* **if**(node.getNodeType() == Node.***ELEMENT\_NODE***) {  
 Element element=(Element) node;  
 System.***out***.println(**"Biro id: "** + element.getAttribute(**"VBiroId"**));  
 System.***out***.println(**"Versenyző idje: "** + element.getAttribute(**"VersenyzoId"**));  
 System.***out***.println(**"Mikor: "** + element.getElementsByTagName(**"Mikor"**).item(0).getTextContent());  
 }  
 }  
 *//Aktuális elem meghatározása* list=document.getElementsByTagName(**"Dij"**);  
  
 **for**(**int** i=0;i<list.getLength();i++){  
 Node node=list.item(i);  
 System.***out***.println(**"\nAktuális elem: "** + node.getNodeName());  
 *//Doksi adatainak kiirása* **if**(node.getNodeType() == Node.***ELEMENT\_NODE***) {  
 Element element=(Element) node;  
 System.***out***.println(**"Dij id: "** + element.getAttribute(**"VBiroId"**));  
 System.***out***.println(**"Értéke: "** + element.getElementsByTagName(**"Erteke"**).item(0).getTextContent());  
 System.***out***.println(**"Át adási időpont: "** + element.getElementsByTagName(**"At\_adasi\_idopont"**).item(0).getTextContent());  
 }  
 }  
  
 *//Aktuális elem meghatározása* list=document.getElementsByTagName(**"Helyszin"**);  
  
 **for**(**int** i=0;i<list.getLength();i++){  
 Node node=list.item(i);  
 System.***out***.println(**"\nAktuális elem: "** + node.getNodeName());  
 *//Doksi adatainak kiirása* **if**(node.getNodeType() == Node.***ELEMENT\_NODE***) {  
 Element element=(Element) node;  
 System.***out***.println(**"Szektor id: "** + element.getAttribute(**"SzektorAzon"**));  
 System.***out***.println(**"Helyszin tipusa: "** + element.getElementsByTagName(**"Helyszin\_tipusa"**).item(0).getTextContent());  
 System.***out***.println(**"Mérete: "** + element.getElementsByTagName(**"Meret"**).item(0).getTextContent());  
 System.***out***.println(**"Szektor: "** + element.getElementsByTagName(**"Szektor"**).item(0).getTextContent());  
 }  
 }  
  
 *//Aktuális elem meghatározása* list=document.getElementsByTagName(**"Hal\_Tipus"**);  
  
 **for**(**int** i=0;i<list.getLength();i++){  
 Node node=list.item(i);  
 System.***out***.println(**"\nAktuális elem: "** + node.getNodeName());  
 *//Doksi adatainak kiirása* **if**(node.getNodeType() == Node.***ELEMENT\_NODE***) {  
 Element element=(Element) node;  
 System.***out***.println(**"Hal\_tipus id: "** + element.getAttribute(**"index"**));  
 System.***out***.println(**"Ragadozó: "** + element.getElementsByTagName(**"Ragadozo"**).item(0).getTextContent());  
 System.***out***.println(**"Tipusa: "** + element.getElementsByTagName(**"Tipusa"**).item(0).getTextContent());  
 System.***out***.println(**"Oshonos: "** + element.getElementsByTagName(**"Oshonos"**).item(0).getTextContent());  
 }  
 }  
  
  
 }**catch** (ParserConfigurationException e){  
 e.printStackTrace();  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 }  
}

## 2.2 DOM Adat módosítás

A Versenyző táblából megkeresi azt a versenyzőt, akinek az id-je 2-es majd a születési helyét módosítja a kódban megadottra majd egy új xml fájlba kiírja a módosított xml fájlt.

package hu.domparse.LWUIJ2;  
  
import org.w3c.dom.Document;  
import org.w3c.dom.Element;  
import org.w3c.dom.Node;  
import org.w3c.dom.NodeList;  
  
import javax.xml.parsers.DocumentBuilder;  
import javax.xml.parsers.DocumentBuilderFactory;  
import javax.xml.parsers.ParserConfigurationException;  
import javax.xml.transform.Result;  
import javax.xml.transform.Source;  
import javax.xml.transform.Transformer;  
import javax.xml.transform.TransformerFactory;  
import javax.xml.transform.dom.DOMSource;  
import javax.xml.transform.stream.StreamResult;  
import java.io.File;  
  
public class DOMModifyLWUIJ2 {  
 public static void main(String[] args) {  
 NodeList nodeList;  
 try{  
 DocumentBuilderFactory factory =DocumentBuilderFactory.newInstance();  
 DocumentBuilder builder=factory.newDocumentBuilder();  
 *//Fájl beolvasása* Document document=builder.parse(new File("XMLLWUIJ2.xml"));  
 document.getDocumentElement().normalize();  
 *//Aktuális elem meghatározása* nodeList=document.getElementsByTagName("Versenyzo");  
 for(int i =0;i<nodeList.getLength();i++){  
 Node node=nodeList.item(i);  
 System.out.println("\nAktuális elem: " + node.getNodeName());  
 *//Versenyzo adatainak kiirása* if(node.getNodeType()==Node.ELEMENT\_NODE){  
 Element element=(Element) node;  
 *//Versenyzo idjének eltárolása* String id= element.getAttribute("VAzon");  
  
 System.out.println("Versenyzo id: " + element.getAttribute("VAzon"));  
 System.out.println("Szektorazonositó id: " + element.getAttribute("Szektorazon"));  
 System.out.println("Díj id: " + element.getAttribute("Dazon"));  
 System.out.println("VersenyzőNév: "  
 + element.getElementsByTagName("VNev").item(0).getTextContent());  
  
 *//Versenyző születési adatainak kiirása* Node nodeszuladat=nodeList.item(i);  
 if(nodeszuladat.getNodeType()==Node.ELEMENT\_NODE){  
 Element elementszuldat=(Element) node;  
 *//id vizsgálata ha megegyezik a megadot id-vel akkor születési hely módositás* if(id.equals("2")) {  
 elementszuldat.getElementsByTagName("Szul\_hely").item(0).setTextContent("Los Angeles");  
 }  
 System.out.println("Születési idő: " + elementszuldat.getElementsByTagName("Szul\_ido").item(0).getTextContent());  
 System.out.println("Születési hely: " + elementszuldat.getElementsByTagName("Szul\_hely").item(0).getTextContent());  
 }  
 }  
 }  
 *//létrehozza az xml fájlt módositva* Transformer transformer= TransformerFactory.newInstance().newTransformer();  
 Source input=new DOMSource(document);  
 Result output=new StreamResult(new File("XMLLWUIJ2Modify.xml"));  
 transformer.transform(input,output);  
 }catch (ParserConfigurationException e){  
 e.printStackTrace();  
 } catch (Exception e){  
 e.printStackTrace();  
 }  
 }  
}

## 2.3 Adatlekérdezés

A hal típus táblából kiírja ki listázza azokat, amelyek védetek, és kiírja a consolera.

package hu.domparse.LWUIJ2;  
  
import org.w3c.dom.Document;  
import org.w3c.dom.Element;  
import org.w3c.dom.Node;  
import org.w3c.dom.NodeList;  
import org.xml.sax.SAXException;  
  
import javax.xml.parsers.DocumentBuilder;  
import javax.xml.parsers.DocumentBuilderFactory;  
import javax.xml.parsers.ParserConfigurationException;  
import java.io.File;  
import java.io.IOException;  
  
public class DOMQueryLWUIJ2 {  
 public static void main(String[] args) {  
 try {  
 NodeList nodeList;  
  
 DocumentBuilderFactory factory =DocumentBuilderFactory.newInstance();  
 DocumentBuilder builder=factory.newDocumentBuilder();  
 *//Fájl beolvasása* Document document=builder.parse(new File("XMLLWUIJ2.xml"));  
 document.getDocumentElement().normalize();  
  
 *//Aktuális elem meghatározása* nodeList=document.getElementsByTagName("Hal\_Tipus");  
  
 for(int i =0;i<nodeList.getLength();i++){  
 Node node=nodeList.item(i);  
  
 *//Hal tipusok adatainak kiirása* if(node.getNodeType()==Node.ELEMENT\_NODE){  
 Element element=(Element) node;  
 String tipus= element.getElementsByTagName("Tipusa").item(0).getTextContent();  
  
 if(tipus.equals("VedettHal"))  
 {  
 System.out.println("\nAktuális elem: " + node.getNodeName());  
 System.out.println("-------");  
 System.out.println("");  
 System.out.println("Hal tipus id: " + element.getAttribute("index"));  
 System.out.println("Ragadozó: " + element.getElementsByTagName("Ragadozo").item(0).getTextContent());  
 System.out.println("Tipusa: " + element.getElementsByTagName("Tipusa").item(0).getTextContent());  
 System.out.println("Oshonos: " + element.getElementsByTagName("Oshonos").item(0).getTextContent());  
  
 }  
 }  
 }  
 } catch (ParserConfigurationException | IOException | SAXException e){  
 e.printStackTrace();  
 }  
 }  
}

## 2.4 DomWrite

package hu.domparse.LWUIJ2;

import org.w3c.dom.\*;

import javax.xml.parsers.DocumentBuilder;

import javax.xml.parsers.DocumentBuilderFactory;

import javax.xml.parsers.ParserConfigurationException;

import javax.xml.transform.\*;

import javax.xml.transform.dom.DOMSource;

import javax.xml.transform.stream.StreamResult;

import java.io.File;

public class DOMWriteLWUIJ2 {

public static void main(String[] args) {

try {

DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();

DocumentBuilder builder = factory.newDocumentBuilder();

Document document = builder.newDocument();

Element root = document.createElement("root");

root.setAttribute("xmlns:xsi", "http://www.w3.org/2001/XMLSchema-instance");

root.setAttribute("xsi:noNamespaceSchemaLocation", "DOMParseLWUIJ2/XMLSchemaLWUIJ2.xsd");

document.appendChild(root);

Element versenyzok = document.createElement("Versenyzok");

root.appendChild(versenyzok);

addVersenyzok(document, versenyzok);

Element doksik = document.createElement("Doksik");

root.appendChild(doksik);

addDoksik(document, doksik);

Element versenyBirok = document.createElement("Verseny\_Birok");

root.appendChild(versenyBirok);

addVersenyBirok(document, versenyBirok);

Element kifogottHalak = document.createElement("Kifogott\_Halak");

root.appendChild(kifogottHalak);

addKifogottHalak(document, kifogottHalak);

Element ellenorzesek = document.createElement("Ellenorzesek");

root.appendChild(ellenorzesek);

addEllenorzesek(document, ellenorzesek);

Element dijak = document.createElement("Dijak");

root.appendChild(dijak);

addDijak(document, dijak);

Element helyszinek = document.createElement("Helyszinek");

root.appendChild(helyszinek);

addHelyszinek(document, helyszinek);

Element halTipusok = document.createElement("Hal\_Tipusok");

root.appendChild(halTipusok);

addHalTipusok(document, halTipusok);

TransformerFactory transformerFactory = TransformerFactory.newInstance();

Transformer transformer = transformerFactory.newTransformer();

transformer.setOutputProperty(OutputKeys.INDENT, "yes");

DOMSource domSource = new DOMSource(document);

StreamResult streamResult = new StreamResult(new File("XMLLWUIJ2out.xml"));

transformer.transform(domSource, streamResult);

System.out.println("Az XML fájl sikeresen létrehozva!");

} catch (ParserConfigurationException | TransformerException e) {

e.printStackTrace();

}

}

private static void addVersenyzok(Document document, Element versenyzok) {

String[][] versenyzoAdatok = {

{"1", "1", "1", "Bence", "1997.01.06", "Gyula"},

{"2", "2", "2", "Akos", "2000.10.15", "Miskolc"},

{"3", "3", "3", "Alex", "1999.03.20", "Budapest"}

};

for (String[] adat : versenyzoAdatok) {

Element versenyzo = document.createElement("Versenyzo");

versenyzo.setAttribute("VAzon", adat[0]);

versenyzo.setAttribute("Szektorazon", adat[1]);

versenyzo.setAttribute("Dazon", adat[2]);

versenyzok.appendChild(versenyzo);

Element vNev = document.createElement("VNev");

vNev.appendChild(document.createTextNode(adat[3]));

versenyzo.appendChild(vNev);

Element szAdatok = document.createElement("Sz\_adatok");

versenyzo.appendChild(szAdatok);

Element szulIdo = document.createElement("Szul\_ido");

szulIdo.appendChild(document.createTextNode(adat[4]));

szAdatok.appendChild(szulIdo);

Element szulHely = document.createElement("Szul\_hely");

szulHely.appendChild(document.createTextNode(adat[5]));

szAdatok.appendChild(szulHely);

}

}

private static void addDoksik(Document document, Element doksik) {

String[] igazolvanyok = {"Taj\_Kartya", "Vezetoi\_Engedely", "Szig\_szam", "Lakcim\_kartya"};

for (int i = 1; i <= igazolvanyok.length; i++) {

Element doksi = document.createElement("Doksi");

doksi.setAttribute("Vazon", String.valueOf(i));

doksik.appendChild(doksi);

Element igazolvany = document.createElement("Igazolvany");

igazolvany.appendChild(document.createTextNode(igazolvanyok[i - 1]));

doksi.appendChild(igazolvany);

}

}

private static void addVersenyBirok(Document document, Element versenyBirok) {

String[][] birok = {

{"1", "Kiss", "Joe", "1980.06.20", "Budapest"},

{"2", "Nagy", "Ferenc", "1985.12.03", "Sopron"},

{"3", "Kovacs", "Janos", "1975.02.10", "Gyor"}

};

for (String[] biro : birok) {

Element versenyBiro = document.createElement("Verseny\_Biro");

versenyBiro.setAttribute("BAzon", biro[0]);

versenyBirok.appendChild(versenyBiro);

Element vNev = document.createElement("VNev");

vNev.appendChild(document.createTextNode(biro[1]));

versenyBiro.appendChild(vNev);

Element kNev = document.createElement("KNev");

kNev.appendChild(document.createTextNode(biro[2]));

versenyBiro.appendChild(kNev);

Element szAdatok = document.createElement("Sz\_adatok");

versenyBiro.appendChild(szAdatok);

Element szulIdo = document.createElement("Szul\_ido");

szulIdo.appendChild(document.createTextNode(biro[3]));

szAdatok.appendChild(szulIdo);

Element szulHely = document.createElement("Szul\_hely");

szulHely.appendChild(document.createTextNode(biro[4]));

szAdatok.appendChild(szulHely);

}

}

private static void addKifogottHalak(Document document, Element kifogottHalak) {

String[][] halak = {

{"1", "1", "1", "Csuka", "5", "14:00"},

{"2", "2", "2", "Ponty", "10", "15:00"},

{"3", "3", "3", "Amur", "8", "17:00"}

};

for (String[] hal : halak) {

Element kifogottHal = document.createElement("Kifogott\_hal");

kifogottHal.setAttribute("Hazon", hal[0]);

kifogottHal.setAttribute("Halindex", hal[1]);

kifogottHal.setAttribute("VerAzon", hal[2]);

kifogottHalak.appendChild(kifogottHal);

Element tHal = document.createElement("THal");

tHal.appendChild(document.createTextNode(hal[3]));

kifogottHal.appendChild(tHal);

Element kg = document.createElement("Kg");

kg.appendChild(document.createTextNode(hal[4]));

kifogottHal.appendChild(kg);

Element ido = document.createElement("ido");

ido.appendChild(document.createTextNode(hal[5]));

kifogottHal.appendChild(ido);

}

}

private static void addEllenorzesek(Document document, Element ellenorzesek) {

String[][] ellenorzesekAdatok = {

{"2", "1", "17:55"},

{"1", "1", "17:50"},

{"3", "3", "17:45"},

{"2", "3", "17:40"},

{"1", "2", "17:35"}

};

for (String[] adat : ellenorzesekAdatok) {

Element ellenorzes = document.createElement("Ellenorzes");

ellenorzes.setAttribute("VersenyzoId", adat[0]);

ellenorzes.setAttribute("VBiroId", adat[1]);

ellenorzesek.appendChild(ellenorzes);

Element mikor = document.createElement("Mikor");

mikor.appendChild(document.createTextNode(adat[2]));

ellenorzes.appendChild(mikor);

}

}

private static void addDijak(Document document, Element dijak) {

String[][] dijakAdatok = {

{"1", "Legtobb\_Hal", "5000", "18:00"},

{"2", "Legnagyobb\_Hal", "5000", "18:30"},

{"3", "Gyoztes", "20000", "19:00"}

};

for (String[] adat : dijakAdatok) {

Element dij = document.createElement("Dij");

dij.setAttribute("Dazon", adat[0]);

dijak.appendChild(dij);

Element dNev = document.createElement("DNev");

dNev.appendChild(document.createTextNode(adat[1]));

dij.appendChild(dNev);

Element erteke = document.createElement("Erteke");

erteke.appendChild(document.createTextNode(adat[2]));

dij.appendChild(erteke);

Element atadasiIdo = document.createElement("At\_adasi\_idopont");

atadasiIdo.appendChild(document.createTextNode(adat[3]));

dij.appendChild(atadasiIdo);

}

}

private static void addHelyszinek(Document document, Element helyszinek) {

String[][] helyszinekAdatok = {

{"1", "Alacsony\_part", "2033", "A"},

{"2", "Magas\_part", "25", "B"},

{"3", "Alacsony\_part", "30", "C"}

};

for (String[] adat : helyszinekAdatok) {

Element helyszin = document.createElement("Helyszin");

helyszin.setAttribute("SzektorAzon", adat[0]);

helyszinek.appendChild(helyszin);

Element helyszinTipusa = document.createElement("Helyszin\_tipusa");

helyszinTipusa.appendChild(document.createTextNode(adat[1]));

helyszin.appendChild(helyszinTipusa);

Element meret = document.createElement("Meret");

meret.appendChild(document.createTextNode(adat[2]));

helyszin.appendChild(meret);

Element szektor = document.createElement("Szektor");

szektor.appendChild(document.createTextNode(adat[3]));

helyszin.appendChild(szektor);

}

}

private static void addHalTipusok(Document document, Element halTipusok) {

String[][] halTipusAdatok = {

{"1", "igen", "VedettHal", "nem"},

{"2", "nem", "VedettHal", "igen"},

{"3", "igen", "NemVedettHal", "igen"}

};

for (String[] adat : halTipusAdatok) {

Element halTipus = document.createElement("Hal\_Tipus");

halTipus.setAttribute("index", adat[0]);

halTipusok.appendChild(halTipus);

Element ragadozo = document.createElement("Ragadozo");

ragadozo.appendChild(document.createTextNode(adat[1]));

halTipus.appendChild(ragadozo);

Element tipusa = document.createElement("Tipusa");

tipusa.appendChild(document.createTextNode(adat[2]));

halTipus.appendChild(tipusa);

Element oshonos = document.createElement("Oshonos");

oshonos.appendChild(document.createTextNode(adat[3]));

halTipus.appendChild(oshonos);

}

}

}