

Jegyzőkönyv

Adatkezelés XML környezetben

Féléves feladat

Képzelt oktatási rendszer adattárolása
és kezelése XML és java környezetben

Készítette: Sirankó Boldizsár
BK2VRM

A feladat leírása

Féléves feladatomban a már Adatbázisrendszerek I c. tárgyra készített beadandómat bővítem ki. Egy fiktív oktatási rendszerről van szó, mely számon tartja a különböző iskolákat, az ott hallgatókat és tanítókat, valamint hogy melyik tárgyat ki hallgatja és oktatja.

Az adatbázisban lévő egyedek, és azok tulajdonságai:

A **diák** egyed *név* és *születési dátum* tulajdonsága magától értetődő.

A *kor* egy születési dátumból számolt származtatott tulajdonság.

A diák *megnyert versenyei* egy többértékű tulajdonság, ami a megnyert versenyek nevét tartalmazza.

(Egy diák értelemszerűen több versenyt is nyerhet, ezért többértékű.)

A *D_id* egy sorszám, ami a diák egyértelmű azonosítására szolgál iskolán belül.

Egy diák egyszerre csak egy iskola jár be.

Egy diák több tárgyat is felvehet, és egy tárgyat is felvehet több diák.

Minden diáknak csak egy osztályfőnöke lehet.

Az **iskola** egyed tulajdonságai *egy évszám* és egy *logikai tulajdonság*, ami azt jelenti, hogy 8 vagy négy osztályos iskoláról van-e szó. Továbbá megtalálható az iskola megnevezése is(*név*).

Az *S_id* egy sorszám, ami az iskola egyértelmű azonosítására szolgál.

Egy iskolába több diák is járhat egyszerre, viszont feltételezzük, hogy tanárok is csak egy iskolában dolgoznak.

A **tárgy** egyed három tulajdonsággal rendelkezik:

O_id, ami egy sorszám, ami a tárgy egyed kulcs tulajdonsága.

A *név* a tárgy nevét tartalmazza, míg a *hossz* az óra hosszát percben megadva.

Egy diák több tárgyat is felvehet, és egy tárgyat is felvehet több diák.

Feltételezzük, hogy egy tárgyat egyszerre csak egy tanár oktat (egy iskolán belül).

A **tanár** egyed tulajdonságai a következők:

T_id, ami a tanár egyed kulcs tulajdonsága (Szintén egy sorszám)

A tanár neve.

Valamint a tanár diplomája, ami egy összetett tulajdonság, ami felépül a következő tulajdonságokból:

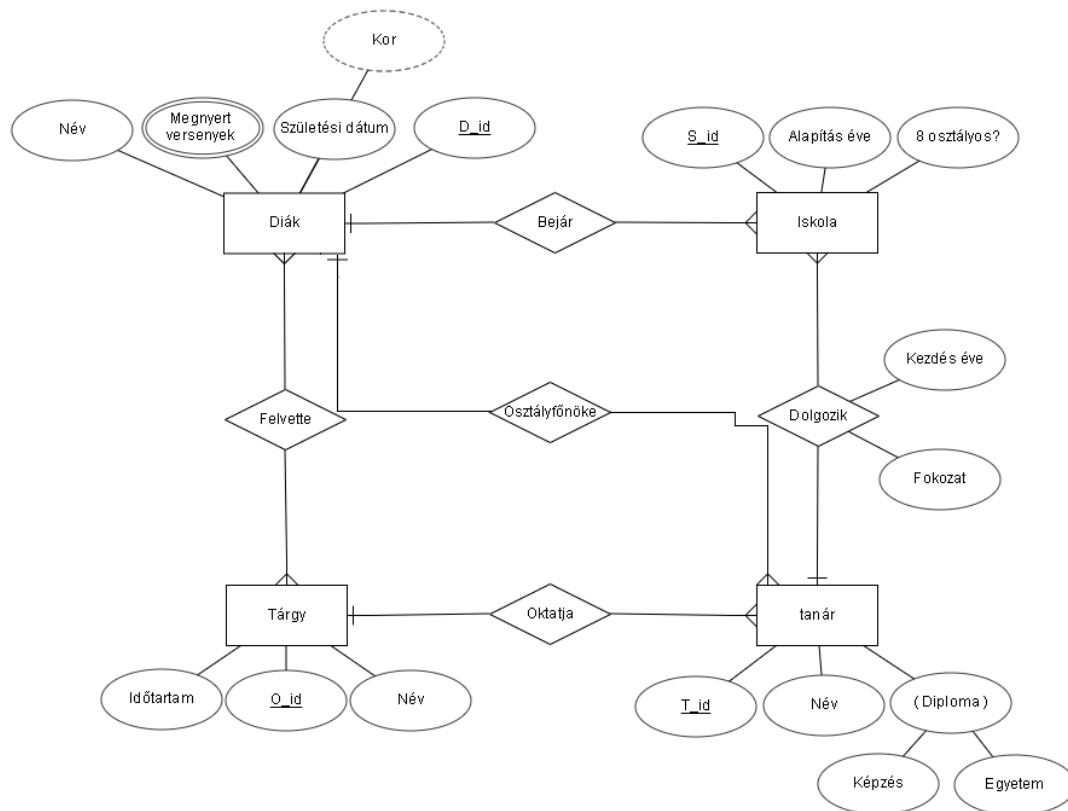
Melyik egyetemen szerezte, és milyen képzésen.

Feltételezzük, hogy egy tárgyat egyszerre csak egy tanár oktat (egy iskolán belül).

Feltételezzük, hogy tanárok csak egy iskolában dolgoznak egyszerre.

Minden diáknak csak egy osztályfőnöke lehet.

Az adatbázis ER modellje:

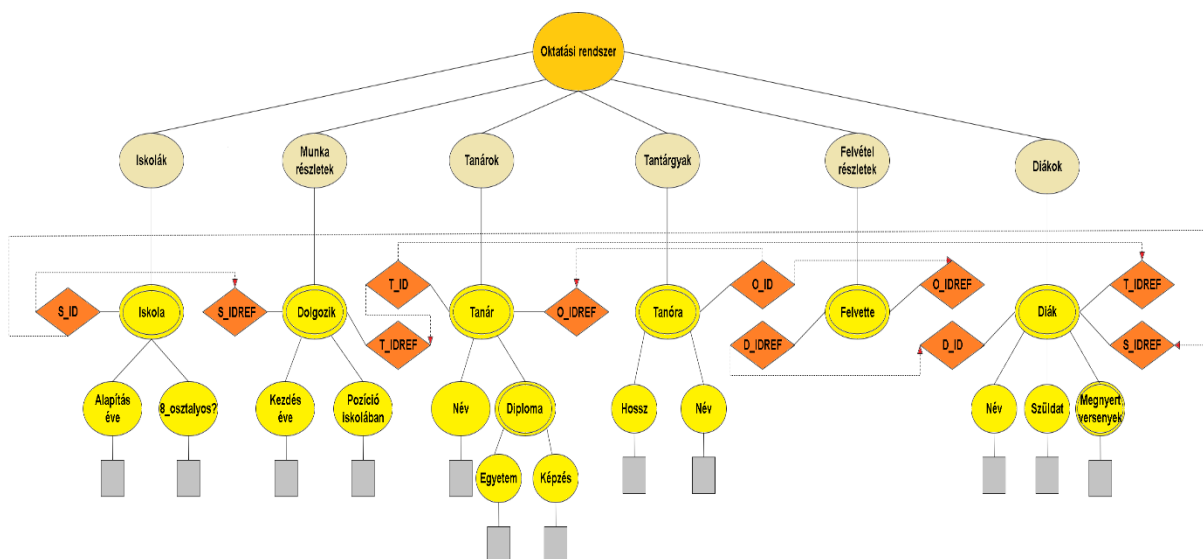


Konvertálás XDM modellre:

Az adatbázis ER modelljének konvertálása eredményezi az idegenkulcsok létrejöttét, S_IDREF, T_IDREF, O_IDREF, D_IDREF, amik referálnak a hozzájuk tartozó elsődleges kulcsra.

Továbbá egy N:M kapcsolatnál kapcsoló elemet hozunk létre, valamint a tanár és az iskola közti kapcsolat tulajdonságainak ábrázolására még egy egyedet vezetünk be, hasonlóan egy 1:N kapcsolathoz.

A kész XDM modell:



XML dokumentum létrehozása az XDM modell alapján:

1-45

```
<?xml version="1.0" encoding="UTF-8"?>
<schoolsystem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="xmlschemaBK2VRM.xsd">

  <schoools>

    <school S_ID="1">
      <name>Academia</name>
      <founded>1735</founded>
      <eightschoolyears>1</eightschoolyears>
    </school>

    <school S_ID="2">
      <name>Tancsics Mihaly Gimnazium</name>
      <founded>1806</founded>
      <eightschoolyears>1</eightschoolyears>
    </school>

    <school S_ID="3">
      <name>Miskolci Iskola</name>
      <founded>1949</founded>
      <eightschoolyears>0</eightschoolyears>
    </school>
  </schoools>

  <lessons>

    <lesson O_ID="1">
      <name>Matematika</name>
      <length>45</length>
    </lesson>

    <lesson O_ID="2">
      <name>XML Adatkezelés</name>
      <length>100</length>
    </lesson>

    <lesson O_ID="3">
      <name>Irodalom</name>
      <length>60</length>
    </lesson>

    <lesson O_ID="4">
      <name>Informatika</name>
      <length>30</length>
    </lesson>
  </lessons>

</schoolsystem>
```

```

46 <teachers>
47   <teacher T_ID="1" O_IDREF="1">
48     <name>Szigeti Tamás</name>
49     <degree>
50       <uni>Szegedi Tudományegyetem</uni>
51       <course>Matematika tanár</course>
52     </degree>
53   </teacher>
54
55   <teacher T_ID="2" O_IDREF="2">
56     <name>Bednarik László</name>
57     <degree>
58       <uni>Debreceni Egyetem</uni>
59       <course>Informatika tanár</course>
60     </degree>
61   </teacher>
62
63   <teacher T_ID="3" O_IDREF="3">
64     <name>Winch Eszter</name>
65     <degree>
66       <uni>Széchenyi Egyetem</uni>
67       <course>Általános filológia</course>
68     </degree>
69   </teacher>
70
71   <teacher T_ID="4" O_IDREF="4">
72     <name>Bitman</name>
73     <degree>
74       <uni>Miskolci Egyetem</uni>
75       <course>Informatika tanár</course>
76     </degree>
77   </teacher>
78
79   <teacher T_ID="5" O_IDREF="1">
80     <name>Pécs Antal</name>
81     <degree>
82       <uni>Selmeci Akadémia</uni>
83       <course>Mérnök</course>
84     </degree>
85   </teacher>
86 </teachers>
87

```

```

88 <jobdetails>
89   <detail S_IDREF="2" T_IDREF="1">
90     <startyear>2000</startyear>
91     <position>Mesteroktató</position>
92   </detail>
93
94   <detail S_IDREF="1" T_IDREF="5">
95     <startyear>1862</startyear>
96     <position>Professzor</position>
97   </detail>
98
99   <detail S_IDREF="3" T_IDREF="2">
100     <startyear>1997</startyear>
101     <position>Mesteroktató</position>
102   </detail>
103
104   <detail S_IDREF="3" T_IDREF="3">
105     <startyear>2020</startyear>
106     <position>Tanársegéd</position>
107   </detail>
108
109   <detail S_IDREF="3" T_IDREF="4">
110     <startyear>1995</startyear>
111     <position>Mesteroktató</position>
112   </detail>
113 </jobdetails>
114
115 <students>
116   <student D_ID="1" S_IDREF="1" T_IDREF="5">
117     <name>Sirankó Boldizsár</name>
118     <birthdate>2000-12-23</birthdate>
119     <woncompetition>Dombó Pál tanulmányi verseny</woncompetition>
120     <woncompetition>Mágocsi tanulmányi verseny</woncompetition>
121     <woncompetition>Nemes Tihamér alkalmazói verseny</woncompetition>
122   </student>
123

```

```

124 <student D_ID="2" S_IDREF="2" T_IDREF="1">
125   <name>Tóth Tóbiás</name>
126   <birthdate>2001-05-04</birthdate>
127   <woncompetition>Nemes Tihamér alkalmazói verseny</woncompetition>
128   <woncompetition>Dusza Árpád programozói verseny</woncompetition>
129
130 </student>
131
132 <student D_ID="3" S_IDREF="3" T_IDREF="2">
133   <name>Kovács Bence</name>
134   <birthdate>1999-05-24</birthdate>
135   <woncompetition>Rubikon történelmi verseny</woncompetition>
136 </student>
137
138 <student D_ID="4" S_IDREF="3" T_IDREF="2">
139   <name>Dékány Péter</name>
140   <birthdate>1998-11-11</birthdate>
141 </student>
142
143 <student D_ID="5" S_IDREF="1" T_IDREF="5">
144   <name>Kovács Antal</name>
145   <birthdate>1999-02-24</birthdate>
146   <woncompetition>Legjobb Firma tehetségkutató verseny</woncompetition>
147 </student>
148 </students>
149
150 <credits>
151   <credit D_IDREF="1" O_IDREF="1"/>
152   <credit D_IDREF="2" O_IDREF="1"/>
153   <credit D_IDREF="3" O_IDREF="2"/>
154   <credit D_IDREF="3" O_IDREF="3"/>
155   <credit D_IDREF="4" O_IDREF="2"/>
156   <credit D_IDREF="4" O_IDREF="3"/>
157   <credit D_IDREF="4" O_IDREF="4"/>
158   <credit D_IDREF="5" O_IDREF="1"/>
159 </credits>
160
161
162 </schoolsystem>

```

XML dokumentum sémájának kialakítása:

1-43

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
3
4     <!--Egyszerű elemek-->
5     <xs:element name="name" type="xs:string"/>
6     <xs:element name="woncompetition" type="xs:string"/>
7     <xs:element name="eightschoolyears" type="xs:boolean"/>
8     <xs:element name="founded" type="xs:integer"/>
9     <xs:element name="length" type="xs:integer"/>
10    <xs:element name="uni" type="xs:string"/>
11    <xs:element name="course" type="xs:string"/>
12    <xs:element name="startyear" type="xs:integer"/>
13    <xs:element name="position" type="xs:string"/>
14    <xs:element name="birthdate" type="xs:date"/>
15
16    <!--Attribútumok-->
17    <xs:attribute name="D_ID" type="xs:integer"/>
18    <xs:attribute name="D_IDREF" type="xs:integer"/>
19    <xs:attribute name="S_ID" type="xs:integer"/>
20    <xs:attribute name="S_IDREF" type="xs:integer"/>
21    <xs:attribute name="T_ID" type="xs:integer"/>
22    <xs:attribute name="T_IDREF" type="xs:integer"/>
23    <xs:attribute name="O_ID" type="xs:integer"/>
24    <xs:attribute name="O_IDREF" type="xs:integer"/>
25
26    <!--Komplex elemek: alapok-->
27
28    <xs:complexType name="degreetype">
29        <xs:sequence>
30            <xs:element ref="uni"/>
31            <xs:element ref="course"/>
32        </xs:sequence>
33    </xs:complexType>
34
35    <xs:complexType name="schooltype">
36        <xs:sequence>
37            <xs:element ref="name"/>
38            <xs:element ref="founded"/>
39            <xs:element ref="eightschoolyears"/>
40        </xs:sequence>
41        <xs:attribute ref="S_ID" use="required"/>
42    </xs:complexType>
43
```



```

44 <xs:complexType name="lessonstype">
45   <xs:sequence>
46     <xs:element ref="name"/>
47     <xs:element ref="length"/>
48   </xs:sequence>
49   <xs:attribute ref="O_ID" use="required"/>
50 </xs:complexType>
51
52 <xs:complexType name="detailtype">
53   <xs:sequence>
54     <xs:element ref="startyear"/>
55     <xs:element ref="position"/>
56   </xs:sequence>
57   <xs:attribute ref="T_IDREF" use="required"/>
58   <xs:attribute ref="S_IDREF" use="required"/>
59 </xs:complexType>
60
61 <xs:complexType name="teachertype">
62   <xs:sequence>
63     <xs:element ref="name"/>
64     <xs:element name="degree" type="degreetype" maxOccurs="1"/>
65   </xs:sequence>
66   <xs:attribute ref="T_ID" use="required"/>
67   <xs:attribute ref="O_IDREF" use="required"/>
68 </xs:complexType>
69
70 <xs:complexType name="studenttype">
71   <xs:sequence>
72     <xs:element ref="name"/>
73     <xs:element ref="birthdate"/>
74     <xs:element ref="woncompetition" maxOccurs="unbounded" minOccurs="0"/>
75   </xs:sequence>
76   <xs:attribute ref="D_ID" use="required"/>
77   <xs:attribute ref="S_IDREF" use="required"/>
78   <xs:attribute ref="T_IDREF" use="required"/>
79 </xs:complexType>
80
81 <xs:complexType name="credittype">
82   <xs:attribute ref="D_IDREF" use="required"/>
83   <xs:attribute ref="O_IDREF" use="required"/>
84 </xs:complexType>
85

```

```

<!--Komplex elemek: gyűjtők-->
<xs:complexType name="lessonstype">
  <xs:sequence>
    <xs:element name="lesson" type="lessonstype" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="schoolstype">
  <xs:sequence>
    <xs:element name="school" type="schooltype" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="teacherstype">
  <xs:sequence>
    <xs:element name="teacher" type="teachertype" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="studentstype">
  <xs:sequence>
    <xs:element name="student" type="studenttype" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="detailstype">
  <xs:sequence>
    <xs:element name="detail" type="detailtype" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="creditstype">
  <xs:sequence>
    <xs:element name="credit" type="creditttype" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

```

```

123 <!--Kész séma-->
124
125 <xs:element name="schoolsystem">
126   <xs:complexType>
127     <xs:sequence>
128       <xs:element name="schools" type="schoolstype"/>
129       <xs:element name="lessons" type="lessonstype"/>
130       <xs:element name="teachers" type="teacherstype"/>
131       <xs:element name="jobdetails" type="detailstype"/>
132       <xs:element name="students" type="studentstype"/>
133       <xs:element name="credits" type="creditstype"/>
134     </xs:sequence>
135   </xs:complexType>
136
137 <!--Elsődleges kulcsok-->
138 <xs:key name="T_ID">
139   <xs:selector xpath="teachers/teacher"/>
140   <xs:field xpath="@T_ID"/>
141 </xs:key>
142
143 <xs:key name="S_ID">
144   <xs:selector xpath="schools/school"/>
145   <xs:field xpath="@S_ID"/>
146 </xs:key>
147
148 <xs:key name="O_ID">
149   <xs:selector xpath="lessons/lesson"/>
150   <xs:field xpath="@O_ID"/>
151 </xs:key>
152
153 <xs:key name="D_ID">
154   <xs:selector xpath="students/student"/>
155   <xs:field xpath="@D_ID"/>
156 </xs:key>
157
158 <!--Idegen kulcsok-->
159 <xs:keyref name="TeacherFK1" refer="T_ID">
160   <xs:selector xpath="students/student"/>
161   <xs:field xpath="@T_IDREF"/>
162 </xs:keyref>
163 <xs:keyref name="TeacherFK2" refer="T_ID">
164   <xs:selector xpath="jobdetails/jobdetail"/>
165   <xs:field xpath="@T_IDREF"/>
166 </xs:keyref>
167

```

```
168 <xs:keyref name="SchoolFK1" refer="S_ID">
169     <xs:selector xpath="students/student"/>
170     <xs:field xpath="@S_IDREF"/>
171 </xs:keyref>
172 <xs:keyref name="SchoolFK2" refer="S_ID">
173     <xs:selector xpath="details/detail"/>
174     <xs:field xpath="@S_IDREF"/>
175 </xs:keyref>
176
177 <xs:keyref name="LessonFK1" refer="O_ID">
178     <xs:selector xpath="teachers/teacher"/>
179     <xs:field xpath="@O_IDREF"/>
180 </xs:keyref>
181 <xs:keyref name="LessonFK2" refer="O_ID">
182     <xs:selector xpath="credits/credit"/>
183     <xs:field xpath="@O_IDREF"/>
184 </xs:keyref>
185
186 <xs:keyref name="StudentFK1" refer="D_ID">
187     <xs:selector xpath="students/student"/>
188     <xs:field xpath="@D_IDREF"/>
189 </xs:keyref>
190 </xs:element>
191 </xs:schema>
```

Java DOM programok:

DOMREAD

```
1  + ...5 lines
6  package hu.domparse.BK2VRM.READ;
7
8  + import ...12 lines
20
21 + /**...4 lines */
25  public class DomReadBK2VRM {
26
27  + /**...3 lines */
30  - public static void main(String[] args) {
31
32      //      Uj DocumentBuilder létrehozasa
33      //      Document létrehozasa XML fajlbol
34      File xmlFile = new File("myxml.xml");
35      Document doc = null;
36      doc = introduceFile(doc, xmlFile);
37
38      //ha sikertelen volt a doksi létrehozasa teszteljuk, egyebkent kiirjuk
39      if (doc != null) {
40          doc.getDocumentElement().normalize();
41          System.out.println("Root element: " + doc.getDocumentElement().getNodeName());
42      } else {
43          System.out.println("A doc null ertek!");
44      }
45      NodeList myList = doc.getDocumentElement().getChildNodes();
46      String indent = "";
47      listData(myList, indent);
48
49  }
```

DOMREAD METÓDUSOK:

introduceFile – File beolvasása

```
public static Document introduceFile(Document doc, File xmlFile) {
    try {
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = factory.newDocumentBuilder();
        doc = dBuilder.parse(xmlFile);

    } catch (ParserConfigurationException | SAXException | IOException ex) {
        Logger.getLogger(DomReadBK2VRM.class.getName()).log(Level.SEVERE, null, ex);
    }
    return doc;
}
```

listData – Adatok kiírása adott NodeListből

```
public static void listData(NodeList thisList, String indent) {
    indent+="\t";
    if (thisList != null) {
        for (int i = 0; i < thisList.getLength(); i++) {
            Node thisNode = thisList.item(i);
            if (thisNode.getNodeType() == Node.ELEMENT_NODE && !thisNode.getTextContent().trim().equals("")) {
                System.out.println(indent+"-"+thisNode.getNodeName()+":");

                NodeList newList = thisNode.getChildNodes();
                listData(newList, indent);
            } else if (thisNode instanceof Text) {
                String value = thisNode.getNodeValue().trim();
                if (value.equals("")) {
                    continue;
                }
                System.out.println(indent+"-"+thisNode.getTextContent()+"-");
            }
        }
    }
}
```

DOMREAD OUTPUT:

```
Root element: schoolsystem
  {schools}:
    {school}:
      {name}:
        -Academia-
      {founded}:
        -1735-
      {eightschoolyears}:
        -1-
    {school}:
      {name}:
        -Tancsics Mihaly Gimnazium-
      {founded}:
        -1806-
      {eightschoolyears}:
        -1-
    {school}:
      {name}:
        -Miskolci Iskola-
      {founded}:
        -1949-
      {eightschoolyears}:
        -0-
  {lessons}:
    {lesson}:
      {name}:
        -Matematika-
      {length}:
        -45-
    {lesson}:
      {name}:
        -XML Adatkezelés-
      {length}:
        -100-
    {lesson}:
      {name}:
        -Irodalom-
      {length}:
        -60-
    {lesson}:
      {name}:
        -Informatika-
      {length}:
        -30-
```

```
38
{teachers}:
  {teacher}:
    {name}:
      -Szigeti Tamás-
    {degree}:
      {uni}:
        -Szegedi Tudományegyetem-
      {course}:
        -Matematika tanár-
  {teacher}:
    {name}:
      -Bednarik László-
    {degree}:
      {uni}:
        -Debreceni Egyetem-
      {course}:
        -Informatika tanár-
  {teacher}:
    {name}:
      -Winch Eszter-
    {degree}:
      {uni}:
        -Széchenyi Egyetem-
      {course}:
        -Általános filológia-
  {teacher}:
    {name}:
      -Bitman-
    {degree}:
      {uni}:
        -Miskolci Egyetem-
      {course}:
        -Informatika tanár-
  {teacher}:
    {name}:
      -Pécs Antal-
    {degree}:
      {uni}:
        -Selmeci Akadémia-
      {course}:
        -Mérnök-
```



```
{jobdetails}:
  {detail}:
    {startyear}:
      -2000-
    {position}:
      -Mesteroktató-
  {detail}:
    {startyear}:
      -1862-
    {position}:
      -Professzor-
  {detail}:
    {startyear}:
      -1997-
    {position}:
      -Mesteroktató-
  {detail}:
    {startyear}:
      -2020-
    {position}:
      -Tanársegéd-
  {detail}:
    {startyear}:
      -1995-
    {position}:
      -Mesteroktató-
{students}:
  {student}:
    {name}:
      -Sirankó Boldizsár-
    {birthdate}:
      -2000-12-23-
    {woncompetition}:
      -Dombó Pál tanulmányi verseny-
    {woncompetition}:
      -Mágocsi tanulmányi verseny-
    {woncompetition}:
      -Nemes Tihamér alkalmazói verseny-
  {student}:
    {name}:
      -Tóth Tóbiás-
    {birthdate}:
      -2001-05-04-
    {woncompetition}:
      -Nemes Tihamér alkalmazói verseny-
    {woncompetition}:
      -Dusza Árpád programozói verseny-
```

```
{student}:
  {name}:
    -Kovács Bence-
  {birthdate}:
    -1999-05-24-
  {woncompetition}:
    -Rubikon történelmi verseny-
{student}:
  {name}:
    -Dékány Péter-
  {birthdate}:
    -1998-11-11-
{student}:
  {name}:
    -Kovács Antal-
  {birthdate}:
    -1999-02-24-
  {woncompetition}:
    -Legjobb Firma tehetségkutató verseny-
```

BUILD SUCCESSFUL (total time: 0 seconds)

DOMMODIFY

Az órák hosszát rövidítjük 10 perccel:

```
1  [+ ...5 lines
6  package hu.domparsing.BK2VRM.MODIFY;
7
8  [+ import ...13 lines
21
22  [+ /**...4 lines */
26  public class DomModifyBK2VRM {
27
28  [+ /**...3 lines */
31  [- public static void main(String[] args) {
32
33      //      Uj DocumentBuilder létrehozasa
34      //      Document létrehozasa XML fajlbol
35      File xmlFile = new File("myxml.xml");
36      Document doc = null;
37      doc = introduceFile(doc, xmlFile);
38
39      //ha sikertelen volt a doksi létrehozasa teszteljuk, egyebkent kiirjuk
40      if (doc != null) {
41          doc.getDocumentElement().normalize();
42          System.out.println("Root element: " + doc.getDocumentElement().getNodeName());
43      } else {
44          System.out.println("A doc null ertek!");
45      }
46      NodeList myList = doc.getDocumentElement().getChildNodes();
47      NodeList modifyList = doc.getDocumentElement().getElementsByTagName("length");
48      //modositas
49      modifyData(modifyList);
50      String indent = "";
51
52      listData(myList, indent);
53
54  }
```

DOMMODIFY METÓDUSOK

introduceFile

listData

modifyData – lesson hosszok rövidítése 10 perccel

```
//orak roviditese 10 perccel
public static void modifyData(NodeList thisList){
    if(thisList!=null){
        for (int i = 0; i < thisList.getLength(); i++) {
            int length = Integer.parseInt(thisList.item(i).getTextContent());
            length-=10;
            Integer myLength = length;
            thisList.item(i).setTextContent(myLength.toString());
        }
    }
}
```

DOMMODIFY OUTPUT

(csak a lényeg)

```
-1-
{school}:
  {name}:
    -Miskolci Iskola-
  {founded}:
    -1949-
  {eightschoolyears}:
    -0-
{lessons}:
  {lesson}:
    {name}:
      -Matematika-
    {length}:
      -35-
  {lesson}:
    {name}:
      -XML Adatkezelés-
    {length}:
      -90-
  {lesson}:
    {name}:
      -Irodalom-
    {length}:
      -50-
  {lesson}:
    {name}:
      -Informatika-
    {length}:
      -20-
{teachers}:
  {teacher}:
    {name}:
      -Szigeti Tamás-
```

DOMQUERY

Azon hallgatók lekérdezése, akik matematikát hallgatnak.

```
1  [+ ...5 lines
6  package hu.domparse.BK2VRM.QUERY;
7
8  [+ import ...16 lines
24
25  [+ /**...4 lines */
29  public class DomQueryBK2VRM {
30
31  [+ /**...3 lines */
34  [- public static void main(String[] args) {
35
36      //      Uj DocumentBuilder létrehozasa
37      //      Document létrehozasa XML fajlbol
38      File xmlFile = new File("myxml.xml");
39      Document doc = null;
40      doc = introduceFile(doc, xmlFile);
41
42      //ha sikertelen volt a doksi létrehozasa teszteljuk, egyebkent kiirjuk
43      if (doc != null) {
44          doc.getDocumentElement().normalize();
45          // System.out.println("Root element: " + doc.getDocumentElement().getNodeName());
46      } else {
47          System.out.println("A doc null ertek!");
48      }
49      ? NodeList myList = doc.getDocumentElement().getChildNodes();
50      //NodeList modifyList = doc.getDocumentElement().getElementsByTagName("length");
51      //modifyData(modifyList);
52      String indent = "";
53
54      //query - matematikat hallgato neve
55      NodeList queryList = doc.getDocumentElement().getElementsByTagName("lesson");
56      //listData(queryList, indent);
57      String keresett = null;
58
59      //Matematika targyak ID-je kikerese
60      for (int i = 0; i < queryList.getLength(); i++) {
61          NodeList query = queryList.item(i).getChildNodes();
62          for (int j = 0; j < query.getLength(); j++) {
63              ? if (query.item(j).getNodeName() == "name" && query.item(j).getTextContent().equals("Matematika")) {
64                  keresett = getParentAttribute(query.item(j), "O_ID");
65              }
66          }
67      }
68
69      //azon hallgato idk kigyujtese akik hallgatnak matematikat
70      ArrayList<String> hallgatoIdk = new ArrayList();
71      NodeList queryList2 = doc.getDocumentElement().getElementsByTagName("credit");
72      for (int i = 0; i < queryList2.getLength(); i++) {
73          if (getAttribute(queryList2.item(i), "O_IDREF").equals(keresett)) {
74              hallgatoIdk.add(getAttribute(queryList2.item(i), "D_IDREF"));
75          }
76      }
77
78
79
80
81      //azon hallgato adatainak kiirasa akik ki lettek gyujtve
82      queryList2 = doc.getDocumentElement().getElementsByTagName("student");
83      for (int i = 0; i < queryList2.getLength(); i++) {
84          for (int j = 0; j < hallgatoIdk.size(); j++) {
85              //System.out.println(getAttribute(queryList2.item(i), "D_ID"));
86              if (getAttribute(queryList2.item(i), "D_ID").equals(hallgatoIdk.get(j))) {
87                  listData(queryList2.item(i).getChildNodes(), indent);
88              }
89          }
90      }
91
92      //indent = "";
93
94      //listData(myList, indent);
95  }
```

DOMQUERY METÓDUSOK

introduceFile

listData

getAttribute, getParentAttribute – jelenlegi node vagy szülő node attribútumjainak kigyűjtése, listázása

```
public static String getAttribute(Node myNode, String ID) {
    NamedNodeMap thisMap = myNode.getAttributes();
    for (int i = 0; i < thisMap.getLength(); i++) {
        if (thisMap.item(i).getNodeName().equals(ID)) {
            return thisMap.item(i).getTextContent();
        }
    }
    return "";
}

public static String getParentAttribute(Node myNode, String ID) {
    NamedNodeMap thisMap = myNode.getParentNode().getAttributes();
    for (int i = 0; i < thisMap.getLength(); i++) {
        if (thisMap.item(i).getNodeName().equals(ID)) {
            return thisMap.item(i).getTextContent();
        }
    }
    return "";
}
```

DOMQUERY OUTPUT

```
run:
  {name}:
    -Sirankó Boldizsár-
  {birthdate}:
    -2000-12-23-
  {woncompetition}:
    -Dombó Pál tanulmányi verseny-
  {woncompetition}:
    -Mágocsi tanulmányi verseny-
  {woncompetition}:
    -Nemes Tihamér alkalmazói verseny-
  {name}:
    -Tóth Tóbiás-
  {birthdate}:
    -2001-05-04-
  {woncompetition}:
    -Nemes Tihamér alkalmazói verseny-
  {woncompetition}:
    -Dusza Árpád programozói verseny-
  {name}:
    -Kovács Antal-
  {birthdate}:
    -1999-02-24-
  {woncompetition}:
    -Legjobb Firma tehetségkutató verseny-
BUILD SUCCESSFUL (total time: 0 seconds)
```

Tartalomjegyzék

A feladat leírása.....	2
Az adatbázisban lévő egyedek, és azok tulajdonságai:.....	2
Az adatbázis ER modellje:	3
Konvertálás XDM modellre:	3
XML dokumentum létrehozása az XDM modell alapján:.....	4
XML dokumentum sémájának kialakítása:	8
Java DOM programok:	13
DOMREAD	13
DOMREAD METÓDUSOK:	14
DOMREAD OUTPUT:	15
DOMMODIFY	19
DOMMODIFY METÓDUSOK.....	19
DOMMODIFY OUTPUT	20
DOMQUERY	21
DOMQUERY METÓDUSOK.....	22
DOMQUERY OUTPUT	23