

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Jogosítvány

Készítette: Krompaszki Patrik
Neptunkód:YJJLDH

Feladat leírása

Egy jogosítvány szerzés adatbázisának az ER-modellezése, átalakítása relációs adatmodellre. Az ER-modell az alábbi jellemzőkkel épül fel:

A **Tanuló** egyedhez tartozik egy Sz. idő tulajdonság egy származtatott tulajdonság az Életkor, az életkor kiszámolható a sz. időből és a mai dátumból, egy összetett tulajdonság a Név, amely két részből áll Vezetéknév és Utónév. A Sz. szám kulcsként funkcionál, mert egyértelműen azonosítja a **Tanulót**.

A **Tanuló** egyedet összeköti a tanul kapcsolat az Autós Iskolával, ez 1:N típusú kapcsolat mivel egy tanuló egy autós iskolába tanul de az autósiskolának többtanulója van. Az **Autósiskola** egyednek van egy kulcstulajdonsága a Név és egy másik tulajdonsága a Helyszín.

Az Autós Iskola egyedet összeköti a k részt vesz kapcsolat a **Kreszvizsga** egyeddel, ez egy N:M kapcsolat mert több **Kreszvizsga** is tartozhat több Autós Iskolához.

A **Kreszvizsga** egyednek több tulajdonsága is van: Eredmény, Időpont, Vizsgadíj. Az Azonosító kulcsként funkcionál.

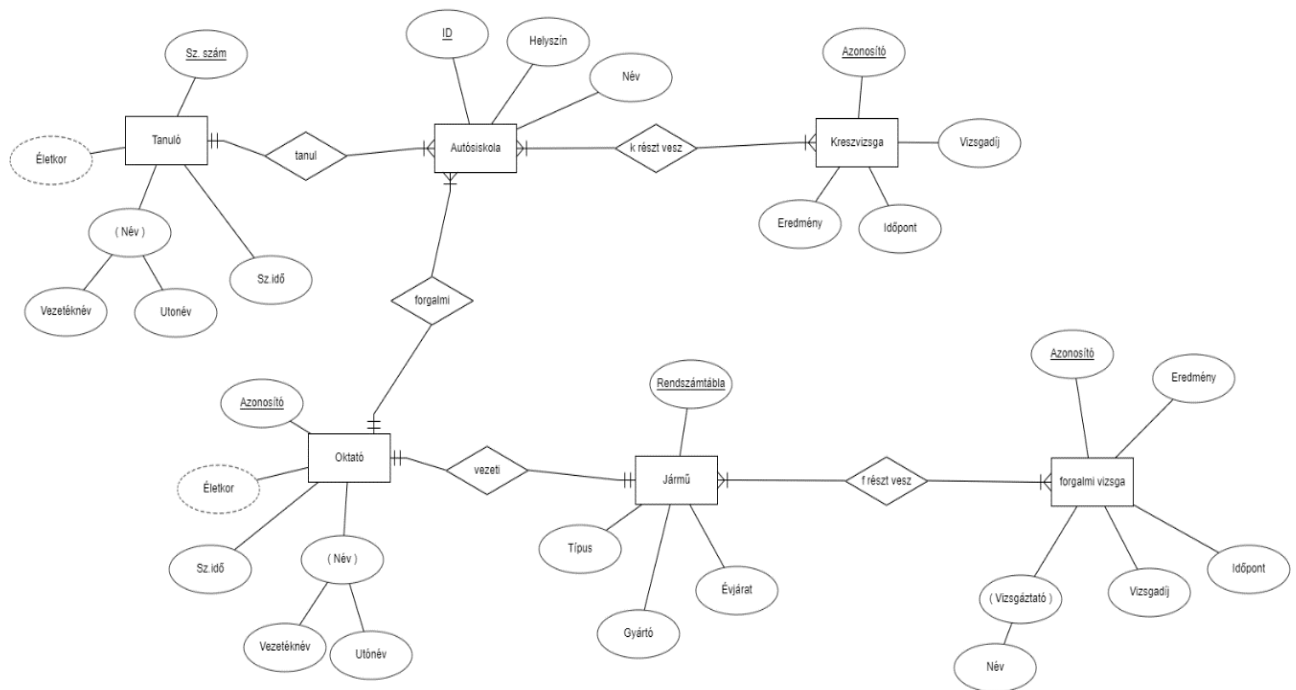
Az **Autóiskola** egyed még kapcsolatban áll az **Oktató** egyeddel, amit egy forgalmi kapcsolat köt össze. Ez egy 1: N kapcsolat mivel általában 1 Oktató egy Autósiskolához és egy Autós iskolának több oktatója is van. Az **Oktatónak** van egy Sz. idő tulajdonsága, van egy összetett tulajdonsága a Név, amelynek két része van Vezetéknév és Utónév, egy származtatott tulajdonság az Életkor, amit a Sz. időből lehet kiszámolni, és egy kulcs tulajdonság, ami az **Azonosító**.

Az **Oktató** egyedet összeköti a vezeti kapcsolat a Jármű egyeddel. Ez egy 1:1 típusú kapcsolat, mivel egy oktatónak egy járműve van és egy oktatóhoz egy jármű tartozik. A jármű egyednek több tulajdonsága is van a Típus, Gyártó és Évjárat. A kulcstulajdonsága a Járműnek a Rendszámtábla, amely egyértelműen megadja a **Járművet**.

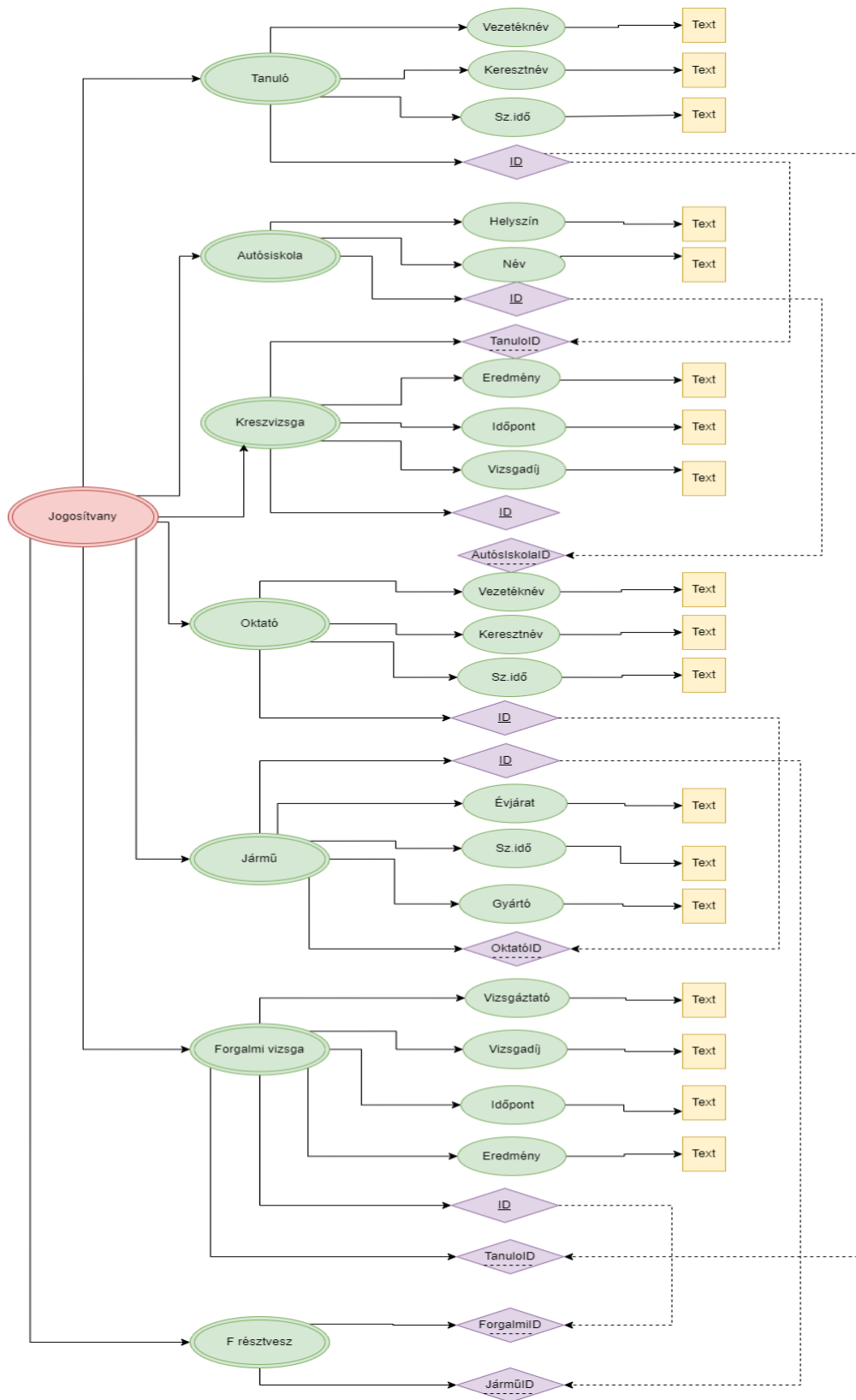
A **Forgalmvizsga** egyedet és a **Jármű** egyedet a f részt vesz kapcsolat köti össze, ez egy 1: N kapcsolat mivel egy jármű van a forgalmi vizsgán, de több forgalmi vizsgához is tartozhat a jármű. A **Jármű** egyednek több tulajdonsága is van Időpont, Eredmény, Vizsgadíj, van egy összetett tulajdonsága a Vizsgáztató és egy része van a Név és az egyedhez tartozik egy többértékű tulajdonság az Útvonal.

1.feladat

1/a ER MODELL



1/b XDM modell



Az XDM modellt a draw.io programmal szerkesztettem. Az XDM modell szerkezetét és kinézetét a World Wide Web Consortium (W3C) [XQuery and XPath Data Model 3.1 –W3C Recommendation](#) című dokumentuma alapján készítettem el. Az egyedekből elemek keletkeztek, az elsődleges és idegen kulcsokból attribútumokká alakultak. A többértékű tulajdonságból egy külön elemet készítünk, ahol a megadható értékeket gyerekelemekként rögzítjük. Az 1:1 és 1: N típusú kapcsolatokat csak idegen kulcsként rögzítjük, az N:M kapcsolatokat pedig kapcsolótáblával kezeljük. Az összetett tulajdonságokat elhagyjuk, csak azokat a tulajdonságokat rögzítjük elemként, amikből az összetett tulajdonság előáll. A származtatott tulajdonságokat is szintén elhagyjuk hiszen azt ki tudjuk mi is számolni a programunkban.

1/c XML Dokumentum

```
1 <?xml version="1.0" encoding="utf-8" standalone="no"?>
2 <jogositvany xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="XSDYJJLDH.xsd">
3   <tanulo id="1">
4     <vezeteknev>Riz</vezeteknev>
5     <keresztnev>Ottó</keresztnev>
6     <szido>1999.10.11</szido>
7   </tanulo>
8   <tanulo id="2">
9     <vezeteknev>Bekre</vezeteknev>
10    <keresztnev>Pál</keresztnev>
11    <szido>1996.07.16</szido>
12  </tanulo>
13  <tanulo id="3">
14    <vezeteknev>Elektrom</vezeteknev>
15    <keresztnev>Ágnes</keresztnev>
16    <szido>2000.12.03</szido>
17  </tanulo>
18  <autosiskola id="1">
19    <nev>UNI Autósiskola</nev>
20    <helyszin>Miskolc</helyszin>
21  </autosiskola>
22  <autosiskola id="2">
23    <nev>Árvai Autósiskola</nev>
24    <helyszin>Miskolc</helyszin>
25  </autosiskola>
26  <kreszvizsga id="1" tanuloID="1">
27    <idopont>2021.11.05</idopont>
28    <eredmeny>Átment</eredmeny>
29    <vizsgadij>4600</vizsgadij>
30  </kreszvizsga>
31  <kreszvizsga id="2" tanuloID="2">
32    <idopont>2021.11.15</idopont>
33    <eredmeny>Mégbukott</eredmeny>
34    <vizsgadij>4600</vizsgadij>
35  </kreszvizsga>
```

```

35 </kreszvizsga>
36 <kreszvizsga id="3" tanuloID="3">
37   <idopont>2021.11.20</idopont>
38   <eredmeny>Átment</eredmeny>
39   <vizsgadij>4600</vizsgadij>
40 </kreszvizsga>
41 <oktato autosiskolaID="1" id="1">
42   <vezeteknev>Meg</vezeteknev>
43   <keresztnev>Győző</keresztnev>
44   <szido>1988.04.16</szido>
45 </oktato>
46 <oktato autosiskolaID="1" id="2">
47   <vezeteknev>Ceruza</vezeteknev>
48   <keresztnev>Elemér</keresztnev>
49   <szido>1975.10.09</szido>
50 </oktato>
51 <oktato autosiskolaID="2" id="3">
52   <vezeteknev>Fehér</vezeteknev>
53   <keresztnev>Farkas</keresztnev>
54   <szido>1979.11.16</szido>
55 </oktato>
56 <oktato autosiskolaID="2" id="4">
57   <vezeteknev>Légy</vezeteknev>
58   <keresztnev>Ottó</keresztnev>
59   <szido>1985.02.16</szido>
60 </oktato>
61 <jarmu id="SSS-567" oktatoID="1">
62   <gyarto>Suzuki</gyarto>
63   <tipus>SX4-CROSS</tipus>
64   <evjarat>2021</evjarat>
65 </jarmu>
66 <jarmu id="MLN-391" oktatoID="2">
67   <gyarto>Skoda</gyarto>
68   <tipus>Octavia</tipus>
69   <evjarat>2015</evjarat>
70 </jarmu>

```

```

70 </jarmu>
71 <jarmu id="PPQ-564" oktatoID="3">
72   <gyarto>Volkswagen</gyarto>
73   <tipus>Passat</tipus>
74   <evjarat>2017</evjarat>
75 </jarmu>
76 <jarmu id="PSW-234" oktatoID="4">
77   <gyarto>Opel</gyarto>
78   <tipus>Astra</tipus>
79   <evjarat>2013</evjarat>
80 </jarmu>
81 <forgalmivizsga id="1" tanuloID="1">
82   <idopont>2021.08.15</idopont>
83   <eredmeny>Átment</eredmeny>
84   <vizsgadij>11000</vizsgadij>
85   <vizsgaztato>Git Áron</vizsgaztato>
86 </forgalmivizsga>
87 <forgalmivizsga id="2" tanuloID="2">
88   <idopont>2021.08.15</idopont>
89   <eredmeny>Megbukott</eredmeny>
90   <vizsgadij>11000</vizsgadij>
91   <vizsgaztato>Git Áron</vizsgaztato>
92 </forgalmivizsga>
93 <forgalmivizsga id="3" tanuloID="3">
94   <idopont>2021.08.20</idopont>
95   <eredmeny>Átment</eredmeny>
96   <vizsgadij>11000</vizsgadij>
97   <vizsgaztato>Trab Antal</vizsgaztato>
98 </forgalmivizsga>
99 </jogositvany>

```

1/d XML Schema

```
1  <?xml version="1.0" encoding="utf-8"?>
2
3  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
4    <xs:element name="jogositvany">
5      <xs:complexType>
6        <xs:sequence>
7          <xs:choice maxOccurs="unbounded">
8            <xs:element name="tanulo" type="tanulotype" />
9            <xs:element name="autosiskola" type="autosiskolatype" />
10           <xs:element name="kreszvizsga" type="kresztype" />
11           <xs:element name="oktato" type="oktatotype" />
12           <xs:element name="jarmu" type="jarmutype" />
13           <xs:element name="forgalmivizsga" type="forgalmitype" />
14         </xs:choice>
15       </xs:sequence>
16     </xs:complexType>
17   </xs:element>
18
19   <!-- egyedileg definiált komplex típusok -->
20   <xs:complexType name="tanulotype">
21     <xs:sequence>
22       <xs:element name="vezeteknev" type="xs:string"/>
23       <xs:element name="keresztnev" type="xs:string"/>
24       <xs:element name="szido" type="xs:string"/>
25     </xs:sequence>
26     <xs:attribute name="id" type="xs:unsignedByte" use="required" />
27   </xs:complexType>
28   <xs:complexType name="autosiskolatype">
29     <xs:sequence>
30       <xs:element name="nev" type="xs:string"/>
31       <xs:element name="helyszin" type="xs:string"/>
32     </xs:sequence>
33     <xs:attribute name="id" type="xs:unsignedByte" use="required" />
34   </xs:complexType>
```

```
18   <!-- key - keyref azonossági megszorítások -->
19   <xs:key name="tanuloID">
20     <xs:selector xpath="./tanulo" />
21     <xs:field xpath="@id" />
22   </xs:key>
23   <xs:keyref name="tanuloIDref" refer="tanuloID">
24     <xs:selector xpath="./kreszvizsga | ./forgalmivizsga" />
25     <xs:field xpath="@tanuloID" />
26   </xs:keyref>
27
28   <xs:key name="autosiskolaID">
29     <xs:selector xpath="./autosiskola" />
30     <xs:field xpath="@id" />
31 </xs:key>
32   <xs:keyref name="autosiskolaIDref" refer="autosiskolaID">
33     <xs:selector xpath="./oktato" />
34     <xs:field xpath="@autosiskolaID" />
35   </xs:keyref>
36
37   <xs:key name="oktatoID">
38     <xs:selector xpath="./oktato" />
39     <xs:field xpath="@id" />
40   </xs:key>
41   <xs:keyref name="oktatoIDref" refer="oktatoID">
42     <xs:selector xpath="./jarmu" />
43     <xs:field xpath="@oktatoID" />
44   </xs:keyref>
45
46 </xs:element>
47
```

```

35 <xs:complexType name="kresztype">
36   <xs:sequence>
37     <xs:element name="idopont" type="xs:string"/>
38     <xs:element name="eredmeny" type="xs:string"/>
39     <xs:element name="vizsgadij" type="xs:string"/>
40   </xs:sequence>
41   <xs:attribute name="id" type="xs:unsignedByte" use="required" />
42   <xs:attribute name="tanuloID" type="xs:unsignedByte" use="required" />
43 </xs:complexType>
44 <xs:complexType name="oktatotype">
45   <xs:sequence>
46     <xs:element name="vezeteknev" type="xs:string"/>
47     <xs:element name="keresztnev" type="xs:string"/>
48     <xs:element name="szido" type="xs:string"/>
49   </xs:sequence>
50   <xs:attribute name="id" type="xs:unsignedByte" use="required" />
51   <xs:attribute name="autosiskolaID" type="xs:unsignedByte" use="required" />
52 </xs:complexType>
53 <xs:complexType name="jarmutype">
54   <xs:sequence>
55     <xs:element name="gyarto" type="xs:string"/>
56     <xs:element name="tipus" type="xs:string"/>
57     <xs:element name="evjarat" type="xs:string"/>
58   </xs:sequence>
59   <xs:attribute name="id" type="carid" use="required" />
60   <xs:attribute name="oktatoID" type="xs:unsignedByte" use="required" />
61 </xs:complexType>
62 <xs:complexType name="forgalmitype">
63   <xs:sequence>
64     <xs:element name="idopont" type="xs:string"/>
65     <xs:element name="eredmeny" type="xs:string"/>
66     <xs:element name="vizsgadij" type="xs:string"/>
67     <xs:element name="vizsgaztato" type="xs:string"/>
68   </xs:sequence>
69   <xs:attribute name="id" type="xs:unsignedByte" use="required" />
70   <xs:attribute name="tanuloID" type="xs:unsignedByte" use="required" />
71 </xs:complexType>

```

```

71 </xs:complexType>
72
73 <!-- egyedileg definiált egyszerű típusok -->
74 <xs:simpleType name="carid">
75   <xs:restriction base="xs:string">
76     <xs:pattern value="[A-Z]{3}-[0-9]{3}" />
77   </xs:restriction>
78 </xs:simpleType>
79
80 </xs:schema>

```


2.Feladat

2/a adatolvasás XML fájl beolvasása XMLJJLDH.xml fájlból és kiíratása a konzolba is megjelenik

```
1 package hu.domparsing.yjjldh;
2
3 import java.io.*;
4
5 public class DomReadYJJLDH {
6     public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException {
7         // XML fájl beolvasása XMLJJLDH.xml fájlból és kiíratása
8         // a konzolba is megjelenik
9         File xml = new File("C:\\xml\\XMLTASKYJJLDH\\XMLYJJLDH.xml");
10
11         // XML fájl DOM document való formába való alakítása
12         DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
13         DocumentBuilder builder = factory.newDocumentBuilder();
14         Document document = builder.parse(xml);
15
16         // DOM document átalakítása DOM DocumentTraversal formába
17         DocumentTraversal traversal = (DocumentTraversal) document;
18
19         // DOM TreeWalker inicializálása
20         // a gyökérelemről kezdve bejárhatjuk az összes elemet és szöveget tartalmazó
21         // csomópontot
22         TreeWalker walker = traversal.createTreeWalker(document.getDocumentElement(),
23             NodeFilter.SHOW_ELEMENT | NodeFilter.SHOW_TEXT, null, true);
24
25         // a DOM bejárása és kiíratása rekurzívan
26         DomTraverser.traverseLevel(walker, "");
27     }
28
29     private static class DomTraverser {
30         public static void traverseLevel(TreeWalker walker, String indent) {
31             // kimentjük az aktuális csomópontot
32             Node node = walker.getCurrentNode();
33
34             // kiíratjuk a megfelelő metódussal
35             if (node.getNodeType() == Node.ELEMENT_NODE) {
36                 printElementNode(node, indent);
37             } else {
38                 printTextNode(node, indent);
39             }
40
41             // rekurzívan meghívjuk a bejárást a DOM fa eggyel mélyebben lévő csomópontjára,
42             // majd azok testvér csomópontjaira
43             for (Node n = walker.firstChild(); n != null; n = walker.nextSibling()) {
44                 traverseLevel(walker, indent + "  ");
45             }
46
47             walker.setCurrentNode(node);
48         }
49
50         private static void printElementNode(Node node, String indent) {
51             System.out.print(indent + node.getNodeName());
52
53             printElementAttributes(node.getAttributes());
54         }
55
56         private static void printElementAttributes(NamedNodeMap attributes) {
57             int length = attributes.getLength();
58
59             if (length > 0) {
60                 System.out.print(" ");
61
62                 for (int i = 0; i < length; i++) {
63                     Node attribute = attributes.item(i);
64
65                     System.out.printf("%s=%s", attribute.getNodeName(), attribute.getNodeValue(),
66                         i != length - 1 ? ", " : "");
67                 }
68             }
69         }
70     }
71 }
```

```

70         i != length - 1 ? ", " : "");
71     }
72
73     System.out.println("");
74 } else {
75     System.out.println();
76 }
77 }
78
79 private static void printTextNode(Node node, String indent) {
80     String content_trimmed = node.getTextContent().trim();
81
82     if (content_trimmed.length() > 0) {
83         System.out.print(indent);
84         System.out.printf("%s\n", content_trimmed);
85     }
86 }
87 }
88
89 }

```

Kimenet

jogositvany (xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance, xsi:noNamespaceSchemaLocation=XSDYJJLDH.xsd)

```
tanulo (id=1)
  vezeteknev
    Riz
  keresztnév
    Ottó
  szido
    1999.10.11
tanulo (id=2)
  vezeteknev
    Bekre
  keresztnév
    Pál
  szido
    1996.07.16
tanulo (id=3)
  vezeteknev
    Elektrom
  keresztnév
    Ágnes
  szido
    2000.12.03
autosiskola (id=1)
  nev
    UNI Autósiskola
  helyszin
    Miskolc
autosiskola (id=2)
  nev
    Árvai Autósiskola
  helyszin
    Miskolc
kreszvizsga (id=1, tanuloID=1)
  idopont
    2021.11.05
  eredmeny
    Átment
```

```
    Átment
vizsgadij
  4600
kreszvizsga (id=2, tanuloID=2)
  idopont
    2021.11.15
  eredmeny
    Megbukott
  vizsgadij
    4600
kreszvizsga (id=3, tanuloID=3)
  idopont
    2021.11.20
  eredmeny
    Átment
  vizsgadij
    4600
oktato (autosiskolaID=1, id=1)
  vezeteknev
    Meg
  keresztnév
    Győző
  szido
    1988.04.16
oktato (autosiskolaID=1, id=2)
  vezeteknev
    Ceruza
  keresztnév
    Elemér
  szido
    1975.10.09
oktato (autosiskolaID=2, id=3)
  vezeteknev
    Fehér
  keresztnév
    Farkas
```

```
jarmu (id=SSS-567, oktatoID=1)
  gyarto
    Suzuki
  tipus
    SX4-CROSS
  evjarat
    2021
jarmu (id=MLN-391, oktatoID=2)
  gyarto
    Skoda
  tipus
    Octavia
  evjarat
    2015
jarmu (id=PPQ-564, oktatoID=3)
  gyarto
    Volkswagen
  tipus
    Passat
  evjarat
    2017
jarmu (id=PSW-234, oktatoID=4)
  gyarto
    Opel
  tipus
    Astra
  evjarat
    2013
forgalmivizsga (id=1, tanuloID=1)
  idopont
    2021.08.15
  eredmeny
    Átment
  vizsgadij
    11000
  vizsgaztato
    Git Áron
forgalmivizsga (id=2, tanuloID=2)
```

Adatlekérdezés

DOMQUERY xml fájl beolvasása után a forgalmi vagy kreszvizsga adatait írja ki amit a felhasználó szeretne a konzolba is megjelenik

```
1 package hu.domparsing.yjjldh;
2
3 import java.io.File;
4
5
6
7 public class DomQueryYJJLDH {
8
9     @SuppressWarnings("resource")
10    public static void main(String[] args){
11
12        //DOMQUERY xml fájl beolvasása után a forgalmi vagy kreszvizsga adatait írja ki amit a felhasználó szeretne
13        //a konzolba is megjelenik
14        try {
15            // XML fájl beolvasása XMLJJLDH.xml fájlból
16            File inputFile = new File("C:\\xml\\XMLTASKYJJLDH\\XMLYJJLDH.xml");
17
18            // DOM doc készítése az xml ből
19            DocumentBuilderFactory dbFactory
20                = DocumentBuilderFactory.newInstance();
21            DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
22            Document doc = dBuilder.parse(inputFile);
23            //gyökér elem
24            doc.getDocumentElement().normalize();
25            System.out.println("Root element : "
26                + doc.getDocumentElement().getNodeName());
27            //melyik vizsgatípus adatait szeretné lekérni
28            System.out.println("Melyik vizsgát szeretné lekérni? forgalmivizsga vagy kreszvizsga");
29            Scanner sc = new Scanner(System.in);
30            String vizsga = sc.nextLine();
31            if( vizsga.equals("forgalmivizsga")) {
32                NodeList nList = doc.getElementsByTagName("forgalmivizsga");
33                for (int temp = 0; temp < nList.getLength(); temp++) {
34                    Node nNode = nList.item(temp);
35                    System.out.println("\nCurrent Element : "
36                        + nNode.getNodeName());
37                    if (nNode.getNodeType() == Node.ELEMENT_NODE) {
38                        Element eElement = (Element) nNode;
39                        System.out.println("Forgalmivizsga ID: "
40                            + eElement.getAttribute("id"));
41                        System.out.println("Időpont : "
42                            + eElement
43                                .getElementsByTagName("idopont")
44                                    .item(0)
45                                    .getTextContent());
46                        System.out.println("Eredmény : "
47                            + eElement
48                                .getElementsByTagName("eredmeny")
49                                    .item(0)
50                                    .getTextContent());
51                        System.out.println("Vizsgadíj : "
52                            + eElement
53                                .getElementsByTagName("fvizsgadij")
54                                    .item(0)
55                                    .getTextContent());
56                        System.out.println("Vizsgáztató : "
57                            + eElement
58                                .getElementsByTagName("vizsgaztato")
59                                    .item(0)
60                                    .getTextContent());
61                    }
62                }
63            }
64            else {
65                NodeList nList = doc.getElementsByTagName("kreszvizsga");
66                for (int temp = 0; temp < nList.getLength(); temp++) {
67                    Node nNode = nList.item(temp);
68                    System.out.println("\nCurrent Element : "
69                        + nNode.getNodeName());
70                }
71            }
72        }
73    }
74}
```

```

79         if (nNode.getNodeType() == Node.ELEMENT_NODE) {
80             Element eElement = (Element) nNode;
81             System.out.println("Kreszvizsga ID: "
82                 + eElement.getAttribute("id"));
83             System.out.println("Időpont : "
84                 + eElement
85                 .getElementsByTagName("idopont")
86                 .item(0)
87                 .getTextContent());
88             System.out.println("Eredmény : "
89                 + eElement
90                 .getElementsByTagName("eredmeny")
91                 .item(0)
92                 .getTextContent());
93             System.out.println("Vizsgadíj : "
94                 + eElement
95                 .getElementsByTagName("vizsgadij")
96                 .item(0)
97                 .getTextContent());
98         }
99     }
100 }
101
102 } catch (Exception e) {
103     e.printStackTrace();
104 }
105
106 }
107 }
108

```

Kimenet

Root element :jogositvany

Melyik vizsgát szeretné lekérni? forgalmivizsga vagy kreszvizsga
kreszvizsga

Current Element :kreszvizsga

Kreszvizsga ID: 1

Időpont : 2021.11.05

Eredmény : Átment

Vizsgadíj : 4600

Current Element :kreszvizsga

Kreszvizsga ID: 2

Időpont : 2021.11.15

Eredmény : Megbukott

Vizsgadíj : 4600

Current Element :kreszvizsga

Kreszvizsga ID: 3

Időpont : 2021.11.20

Eredmény : Átment

Vizsgadíj : 4600

Root element :jogositvany

Melyik vizsgát szeretné lekérni? forgalmivizsga vagy kreszvizsga
forgalmivizsga

Current Element :forgalmivizsga

Forgalmivizsga ID: 1

Időpont : 2021.08.15

Eredmény : Átment

Vizsgadíj : 11000

Vizsgáztató : Git Áron

Current Element :forgalmivizsga

Forgalmivizsga ID: 2

Időpont : 2021.08.15

Eredmény : Megbukott

Vizsgadíj : 11000

Vizsgáztató : Git Áron

Current Element :forgalmivizsga

Forgalmivizsga ID: 3

Időpont : 2021.08.20

Eredmény : Átment

Vizsgadíj : 11000

Vizsgáztató : Trab Antal

2/c Módosítás

//DomModify xml fájl beolvasása után 2 adat kerül módosításra új xml fájl keletkezik és a konzolba is megjelenik

```
1 package hu.domparse.yjjldh;
2
3 import java.io.*;
4
17 public class DomModifyYJJLDH {
18     public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException,
19         XPathExpressionException, DOMException, ParseException, TransformerException {
20
21         //DomModify xml fájl beolvasása után 2 adat kerül módosításra
22         //Új xml fájl keletkezik és a konzolba is megjelenik
23
24         // XML fájl létrehozása és beolvasása XMLJJLDH.xml fájlból
25         File xml = new File("C:\\xml\\XMLTASKYJJLDH\\XMLYJJLDH.xml");
26
27         // DOM doc készítése az xmlből
28         DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
29         DocumentBuilder builder = factory.newDocumentBuilder();
30         Document document = builder.parse(xml);
31
32         // a DOM doc adatainak módosítása
33         DomModifier.modifyDom(document);
34
35         // DOM Traversal készítése a DOM Documentből
36         DocumentTraversal traversal = (DocumentTraversal) document;
37
38         // DOM TreeWalker inicializálása
39         TreeWalker walker = traversal.createTreeWalker(document.getDocumentElement(),
40             NodeFilter.SHOW_ELEMENT | NodeFilter.SHOW_TEXT, null, true);
41
42         // rekurzív DOM bejárás
43         DomTraverser.traverseLevel(walker, "");
44
45         //módosított XML létrehozása XMLyjjldh.updated.xml fájlként
46         TransformerFactory transformerFactory = TransformerFactory.newInstance();
47         Transformer transformer = transformerFactory.newTransformer();
48         DOMSource source = new DOMSource(document);
49         StreamResult result = new StreamResult(new File("XMLyjjldh.updated.xml"));
50         transformer.transform(source, result);
51     }
52 }
```

```

private static class DomModifier {
    public static void modifyDom(Document document) throws XPathExpressionException, DOMException, ParseException {
        XPathFactory factory = XPathFactory.newInstance();
        XPath xpath = factory.newXPath();

        // 1.) Az egyik Volkswagen Passatot tanulócsit lecserélték Golfra
        // XPath segítségével lekérdezzük a megfelelő elemet/csomópontot a DOM
        // dokumentumból
        Node fajta = (Node) xpath.evaluate("//jarmu[./gyarto='Volkswagen']/tipus",
            document, XPathConstants.NODE);

        fajta.setTextContent("Golf");

        // 2.) Mivel az egyik tanulo megbukott a kreszvizsgan így újra be kell fizetnie a vizsgadíjat
        NodeList kreszvizsga = (NodeList) xpath.evaluate("//kreszvizsga[./eredmeny='Megbukott']/vizsgadij", document,
            XPathConstants.NODESET);

        for (int i = 0; i < kreszvizsga.getLength(); i++) {
            Node dij = kreszvizsga.item(i);

            String vizsgadij = (dij.getTextContent());
            int vizsgadij1 = Integer.parseInt(vizsgadij);
            dij.setTextContent(Double.toString(vizsgadij1 + 4600));
        }
    }
}

private static class DomTraverser {
    public static void traverseLevel(TreeWalker walker, String indent) {
        // aktuális csomópont
        Node node = walker.getCurrentNode();

        // kiíratás
        if (node.getNodeType() == Node.ELEMENT_NODE) {
            printElementNode(node, indent);
        } else {
            printTextNode(node, indent);
        }

        // rekurzívan meghívjuk a bejárást a DOM fa eggyel mélyebben lévő csomópontjára,
        // majd azok testvér csomópontjaira
        for (Node n = walker.firstChild(); n != null; n = walker.nextSibling()) {
            traverseLevel(walker, indent + "  ");
        }

        walker.setCurrentNode(node);
    }

    private static void printElementNode(Node node, String indent) {
        System.out.print(indent + node.getNodeName());

        printElementAttributes(node.getAttributes());
    }

    private static void printElementAttributes(NamedNodeMap attributes) {
        int length = attributes.getLength();

        if (length > 0) {
            System.out.print(" [ ");

            for (int i = 0; i < length; i++) {
                Node attribute = attributes.item(i);

                System.out.printf("%s=%s", attribute.getNodeName(), attribute.getNodeValue(),
                    i != length - 1 ? ", " : "");
            }
        }
    }
}

```

```
118
119         System.out.println(" ]");
120     } else {
121         System.out.println();
122     }
123 }
124
125 private static void printTextNode(Node node, String indent) {
126     String content_trimmed = node.getTextContent().trim();
127
128     if (content_trimmed.length() > 0) {
129         System.out.print(indent);
130         System.out.printf("{ %s }%n", content_trimmed);
131     }
132 }
133 }
134
135 }
```

Kimenet

```
jogositvany [ xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance, xsi:noNamespaceSchemaLocation=XSDYJJLDH.xsd ]
  tanulo [ id=1 ]
    vezeteknev
      { Riz }
    keresztnév
      { Ottó }
    szido
      { 1999.10.11 }
  tanulo [ id=2 ]
    vezeteknev
      { Bekre }
    keresztnév
      { Pál }
    szido
      { 1996.07.16 }
  tanulo [ id=3 ]
    vezeteknev
      { Elektrom }
    keresztnév
      { Ágnes }
    szido
      { 2000.12.03 }
  autosiskola [ id=1 ]
    nev
      { UNI Autósiskola }
    helyszin
      { Miskolc }
  autosiskola [ id=2 ]
    nev
      { Árvai Autósiskola }
    helyszin
      { Miskolc }
  kreszvizsga [ id=1, tanuloID=1 ]
    idopont
      { 2021.11.05 }
    eredmény
      { Átment }

    vizsgadij
      { 4600 }
  kreszvizsga [ id=2, tanuloID=2 ]
    idopont
      { 2021.11.15 }
    eredmény
      { Megbukott }
    vizsgadij
      { 9200.0 }
  kreszvizsga [ id=3, tanuloID=3 ]
    idopont
      { 2021.11.20 }
    eredmény
      { Átment }
    vizsgadij
      { 4600 }
  oktato [ autosiskolaID=1, id=1 ]
    vezeteknev
      { Meg }
    keresztnév
      { Győző }
    szido
      { 1988.04.16 }
  oktato [ autosiskolaID=1, id=2 ]
    vezeteknev
      { Ceruza }
    keresztnév
      { Elemér }
    szido
      { 1975.10.09 }
  oktato [ autosiskolaID=2, id=3 ]
    vezeteknev
      { Fehér }
    keresztnév
      { Farkas }
    szido
      { 1979.11.16 }

    { 1979.11.16 }
  oktato [ autosiskolaID=2, id=4 ]
    vezeteknev
      { Légy }
    keresztnév
      { Ottó }
    szido
      { 1985.02.16 }
  jarmu [ id=SSS-567, oktatoID=1 ]
    gyarto
      { Suzuki }
    tipus
      { SX4-CROSS }
    evjarat
      { 2021 }
  jarmu [ id=MLN-391, oktatoID=2 ]
    gyarto
      { Skoda }
    tipus
      { Octavia }
    evjarat
      { 2015 }
  jarmu [ id=PPQ-564, oktatoID=3 ]
    gyarto
      { Volkswagen }
    tipus
      { Golf }
    evjarat
      { 2017 }
  jarmu [ id=PSW-234, oktatoID=4 ]
    gyarto
      { Opel }
    tipus
      { Astra }
    evjarat
      { 2013 }
  forgalmivizsga [ id=1, tanuloID=1 ]
```