JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Repülőgép járat nyilvántartó adatbázis

Készítette: **Simkó Levente**

Neptunkód: **YG5YAO**

Dátum: **2023. 12. 10.**

**Tartalomjegyzék**

**A Feladat leírása:2**

**1.Feladat3**

1a) Az adatbázis ER modellje3

1b) Konvertálás XDM modellre 4

1c) XML az XDM alapján 5

1d) XMLSchema XML alapján 7

**2.Feladat11**

2a) Adatolvasás 11

2b) Adatmódosítás 14

2c) Adat lekérdezés 17

2d) Adatírás 19

**A feladat leírása:**

A feladatban egy repülőgép, járat, valamint a hozzájuk tartozó adatok, mint a légitársaság és a repülőgép típusát és a járaton utazó utasokat hogyan lehetne eltárolni egy XML adatbázisban. Egy repülőgépnek elképesztően sok adatát lehetne eltárolni, mint például a repülőgép típusa, melyik légitársasághoz tartozik, milyen járathoz kapcsolódik stb. De vegyük például a légitárságot, amihez kötődik a repülőgép így ezáltal a járat is, de egy társaságnak egy adott típusú repülőből több is lehet, amit repülőgép flottának nevezünk. Az utast is nyilvántartjuk, aki az adott járaton utazik.

De most nézzük meg az egészet részletesebben.

Kezdjük a Légitársasággal. Minden légitársaságot valamilyen módon azonosítani kell, erre a való életben több lehetőség is van, én ebből az IATA (magyarul: Nemzetközi Légi Szállítási Szövetség) kódot használtam fel és tároltam el, mint ID. Emelet eltárolásra került a légitársaság neve, weboldalának címe és az elérhetőségei.

Következő a Repülőgép típusa, ahol ID-ként a típus közismert rövidítését tároltam el, valamint gyártóját és a rendes nevét, illetve a hajtóművet.

Repülőgép esetén ID a légitársaság AITA kódja plusz két karakteres számkombinációból áll a következő formátumban (AITA\_XY). Valamint a repülőgép tartozik valamelyik légitársasághoz, illetve van típusa, így azok is eltárolásra kerültek. Ezen felül eltárolt adatok még, a gyártási szám, hívószám (vagy másképp hívójel) és a személyzet (típusa, például van-e utaskísérő a gépen mert hát pilóta nélkül nem megy a gép, de az is a személyzet része).

Járat esetén ID csak egy J\_XY kombináció, ugyanis a repülőgép ID-ját is eltároljuk, ahonnan minden mást is ki tudunk nyerni már. Ezen felül eltárolásra kerül, hogy honnan indul az adott járat és hova érkezik, valamint az időpont. Az, hogy honnan hova megy szintén a IATA kód alapján van eltárolva, ami repterek esetén három betűből áll össze. (Például LHR az London Heathrow-t jelöl.)

Utasnak nem sokban különbözik az eltárolt adata, mint a légitárságnak, csak weboldal helyett a poggyászt tároljuk el, hogy mekkora és mennyi, valamint a járat ID-ját, amin utazik. ID-nek csak sima 3 számból készült kombinációt adtam meg.

Az általam fentebb felsorolt adatokat vettem fel az adatbázisba.

**1.Feladat**

**1a) Az adatbázis ER modellje:**

Lentebb látható lesz, hogy a repülőgépen van a legnagyobb hangsúly, az mindennek a közepe. Egy légitársaság rendelkezik több repülővel, így ott 1:N kapcsolat van. A repülő meghatározza annak típusát így a repülőgép és repülőgép típus között 1:1 kapcsolat van. Több repülőgép típus lehet több légitársaságnak, amit flottának nevezünk így ott N:M kapcsolat van és a tulajdonság a darabszám. Egy repülőgép csak egy járathoz lehet rendelve így ott 1:1 kapcsolat van. Járaton viszont több utas utazik, így ott 1:N kapcsolat van.

A képen origami, diagram, fehér, minta látható

Automatikusan generált leírás

**1b) Konvertálás XDM modellre:**

**A képen fehér, diagram, sor látható

Automatikusan generált leírás**

Az ER modellt konvertáltam XDM-re. Az Egyedekből Element-ek lettek és ezek az Element-ek többszörös előfordulásúak.

**1c) XML az XDM alapján:**

Az XDM alapján létrehoztam az XML fájlt Visual Studio Code segítségével.

<?xml version="1.0" encoding="UTF-8"?>

<?xml-model href="XMLSchemaYG5YAO.xsd" type="application/xml" schematypens="http://www.w3.org/2001/XMLSchema"?>

<root>

    <!--Lt Lufthansa egyed-->

    <Légitársaság LTársaságID="LH">

        <Név>Deutsche Lufthansa AG</Név>

        <Weboldal>lufthansa.com</Weboldal>

        <Elérhetöség>

            <E-mailcím>contactCustomer@lh.com</E-mailcím>

            <Telefonszám>+361-429-2200</Telefonszám>

        </Elérhetöség>

    </Légitársaság>

    <!--Lt British Airways egyed-->

     <Légitársaság LTársaságID="BA">

        <Név>British Airways</Név>

        <Weboldal>britishairways.com</Weboldal>

        <Elérhetöség>

            <E-mailcím>customerContact@ba.com</E-mailcím>

            <Telefonszám>+361-777-4747</Telefonszám>

        </Elérhetöség>

    </Légitársaság>

    <!--Lt American Airlines egyed-->

     <Légitársaság LTársaságID="AA">

        <Név>American Airlines, Inc.</Név>

        <Weboldal>www.aa.com</Weboldal>

        <Elérhetöség>

            <E-mailcím>customerHelp@aa.com</E-mailcím>

            <Telefonszám>+361-429-2346</Telefonszám>

        </Elérhetöség>

    </Légitársaság>

    <!--RT Airbus 380 egyed-->

    <Repülö\_Tipus RTipusID="A380">

        <Gyartó>Airbus</Gyartó>

        <Elnevezés>Airbus A380</Elnevezés>

        <Hajtómű>4x Rolls-Royce Trent 900</Hajtómű>

    </Repülö\_Tipus>

    <!--RT Boeing 777 egyed-->

    <Repülö\_Tipus RTipusID="B777">

        <Gyartó>Boeing Commercial Airplanes</Gyartó>

        <Elnevezés>Boeing 777</Elnevezés>

        <Hajtómű>2x General Electric GE90</Hajtómű>

    </Repülö\_Tipus>

    <!--RT Boeing 787 Dreamliner-->

    <Repülö\_Tipus RTipusID="B787D">

        <Gyartó>Boeing Commercial Airplanes</Gyartó>

        <Elnevezés>Boeing 787 Dreamliner</Elnevezés>

        <Hajtómű>2x Rolls-Royce Trent 1000</Hajtómű>

    </Repülö\_Tipus>

    <!--F LH  B787D egyed-->

    <Flotta RTipusID="B787D" LTársaságID="LH">

        <Darabszám>358</Darabszám>

    </Flotta>

    <!--F AA  A380 egyed-->

    <Flotta RTipusID="A380" LTársaságID="AA">

        <Darabszám>102</Darabszám>

    </Flotta>

     <!--F BA  B777 egyed-->

     <Flotta RTipusID="B777" LTársaságID="BA">

        <Darabszám>246</Darabszám>

    </Flotta>

    <!--Repülö egyedek-->

    <Repülö RepülöID="AA\_01" RTipusID="A380" LTársaságID="AA">

        <Gyártásiszám>A380-122-449</Gyártásiszám>

        <Hívószám>AA-01</Hívószám>

        <Személyzet>Piloták</Személyzet>

        <Személyzet>Légiutaskisérök</Személyzet>

    </Repülö>

    <Repülö RepülöID="LH\_02" RTipusID="B787D" LTársaságID="LH">

        <Gyártásiszám>B787-122-001</Gyártásiszám>

        <Hívószám>LH-02</Hívószám>

        <Személyzet>Piloták</Személyzet>

        <Személyzet>Légiutaskisérök</Személyzet>

    </Repülö>

    <Repülö RepülöID="BA\_03" RTipusID="B777" LTársaságID="BA">

        <Gyártásiszám>B777-287-019</Gyártásiszám>

        <Hívószám>BA-03</Hívószám>

        <Személyzet>Piloták</Személyzet>

        <Személyzet>Légiutaskisérök</Személyzet>

    </Repülö>

    <!--Járatok-->

    <Járat JáratID="J\_01" RepülöID="BA\_03">

        <Indulás>LHR</Indulás>

        <Cél>BUD</Cél>

        <Idöpont>2024.01.12</Idöpont>

    </Járat>

    <Járat JáratID="J\_02" RepülöID="LH\_02">

        <Indulás>BER</Indulás>

        <Cél>BUD</Cél>

        <Idöpont>2024.02.09</Idöpont>

    </Járat>

    <Járat JáratID="J\_03" RepülöID="AA\_01">

        <Indulás>LAX</Indulás>

        <Cél>LHR</Cél>

        <Idöpont>2024.01.24</Idöpont>

    </Járat>

    <!--Utasok-->

    <Utas UtasID="003" JáratID="J\_01">

        <Név>Simkó Levente</Név>

        <Elérhetöség>

            <E-mailcím>smlevente@gmail.com</E-mailcím>

            <Telefonszám>+3670-429-5555</Telefonszám>

        </Elérhetöség>

        <Poggyász>2db kézi</Poggyász>

    </Utas>

    <Utas UtasID="012" JáratID="J\_03">

        <Név>Magyar Márk</Név>

        <Elérhetöség>

            <E-mailcím>mmark1996@outlook.com</E-mailcím>

            <Telefonszám>+3630-479-8888</Telefonszám>

        </Elérhetöség>

        <Poggyász>1db nagy</Poggyász>

    </Utas>

    <Utas UtasID="048" JáratID="J\_02">

        <Név>Kiss Szabina</Név>

        <Elérhetöség>

            <E-mailcím>szabinakiss14@gmail.com</E-mailcím>

            <Telefonszám>+3620-888-2200</Telefonszám>

        </Elérhetöség>

        <Poggyász>1db kézi</Poggyász>

    </Utas>

</root>

**1d) XMLSchema XML alapján:**

Az XSD is Visual Studio Code segítségével csináltam, ezt saját kezűleg csináltam.

<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"

    xmlns:tns="http://www.example.org/XMLShemaYG5YAO"

    elementFormDefault="qualified">

    <xs:element name="root">

        <xs:complexType>

            <xs:sequence>

                <!--Légitársaság meghatározás-->

                <xs:element name="Légitársaság" minOccurs="1" maxOccurs="unbounded">

                    <xs:complexType>

                        <xs:sequence>

                            <xs:element name="Név" type="xs:string"/>

                            <xs:element name="Weboldal" type="xs:string"/>

                            <xs:element name="Elérhetöség">

                                            <xs:complexType>

                                                <xs:sequence>

                                                    <xs:element name="E-mailcím" type="emailTipus" />

                                                    <xs:element name="Telefonszám" type="telefonTipus" />

                                                </xs:sequence>

                                            </xs:complexType>

                                        </xs:element>

                        </xs:sequence>

                        <xs:attribute name="LTársaságID" type="xs:string" />

                    </xs:complexType>

                </xs:element>

                <!--Repülö\_Tipus meghatározás-->

                <xs:element name="Repülö\_Tipus" minOccurs="1" maxOccurs="unbounded">

                    <xs:complexType>

                        <xs:sequence>

                            <xs:element name="Gyartó" type="xs:string"/>

                            <xs:element name="Elnevezés" type="xs:string"/>

                            <xs:element name="Hajtómű" type="xs:string"/>

                        </xs:sequence>

                        <xs:attribute name="RTipusID" type="xs:string" />

                    </xs:complexType>

                </xs:element>

                <!--Flotta meghatározás-->

                <xs:element name="Flotta" minOccurs="1" maxOccurs="unbounded">

                    <xs:complexType>

                        <xs:sequence>

                            <xs:element name="Darabszám" type="xs:integer"/>

                        </xs:sequence>

                        <xs:attribute name="LTársaságID" type="xs:string" />

                        <xs:attribute name="RTipusID" type="xs:string" />

                    </xs:complexType>

                </xs:element>

                <!--Repülö meghatározás-->

                <xs:element name="Repülö" minOccurs="1" maxOccurs="unbounded">

                    <xs:complexType>

                        <xs:sequence>

                            <xs:element name="Gyártásiszám" type="xs:string"/>

                            <xs:element name="Hívószám" type="xs:string"/>

                            <xs:element name="Személyzet" type="xs:string" minOccurs="1" maxOccurs="unbounded"/>

                        </xs:sequence>

                        <xs:attribute name="RepülöID" type="xs:string" />

                        <xs:attribute name="LTársaságID" type="xs:string" />

                        <xs:attribute name="RTipusID" type="xs:string" />

                    </xs:complexType>

                </xs:element>

                <!--Járat meghatározás-->

                <xs:element name="Járat" minOccurs="1" maxOccurs="unbounded">

                    <xs:complexType>

                        <xs:sequence>

                            <xs:element name="Indulás" type="indulasCelTipus"/>

                            <xs:element name="Cél" type="indulasCelTipus"/>

                            <xs:element name="Idöpont" type="idopontTipus"/>

                        </xs:sequence>

                        <xs:attribute name="JáratID" type="xs:string" />

                        <xs:attribute name="RepülöID" type="xs:string" />

                    </xs:complexType>

                </xs:element>

                <!--Utas meghatározás-->

                <xs:element name="Utas" minOccurs="1" maxOccurs="unbounded">

                    <xs:complexType>

                        <xs:sequence>

                            <xs:element name="Név" type="xs:string"/>

                            <xs:element name="Elérhetöség">

                                            <xs:complexType>

                                                <xs:sequence>

                                                    <xs:element name="E-mailcím" type="emailTipus" />

                                                    <xs:element name="Telefonszám" type="telefonTipus" />

                                                </xs:sequence>

                                            </xs:complexType>

                                        </xs:element>

                            <xs:element name="Poggyász" type="xs:string"/>

                        </xs:sequence>

                        <xs:attribute name="UtasID" type="xs:string" />

                        <xs:attribute name="JáratID" type="xs:string" />

                    </xs:complexType>

                </xs:element>

            </xs:sequence>

        </xs:complexType>

        <!--PK meghatározás egyedeknek-->

        <xs:key name="Key\_legitarsasag">

            <xs:selector xpath="Légitársaság" />

            <xs:field xpath="@LTársaságID" />

        </xs:key>

        <xs:key name="Key\_rtipus">

            <xs:selector xpath="Repülö\_Tipus" />

            <xs:field xpath="@RTipusID" />

        </xs:key>

        <xs:key name="Key\_repulo">

            <xs:selector xpath="Repülö" />

            <xs:field xpath="@RepülöID" />

        </xs:key>

        <xs:key name="Key\_jarat">

            <xs:selector xpath="Járat" />

            <xs:field xpath="@JáratID" />

        </xs:key>

        <xs:key name="Key\_utas">

            <xs:selector xpath="Utas" />

            <xs:field xpath="@UtasID" />

        </xs:key>

        <!--FK meghatározás egyedeknek-->

        <xs:keyref name="FKey\_flotta\_legitarsasag"

            refer="Key\_legitarsasag">

            <xs:selector xpath="Flotta" />

            <xs:field xpath="@LTársaságID" />

        </xs:keyref>

        <xs:keyref name="FKey\_flotta\_repulo\_tipus"

            refer="Key\_rtipus">

            <xs:selector xpath="Flotta" />

            <xs:field xpath="@RTipusID" />

        </xs:keyref>

        <xs:keyref name="FKey\_repulo\_legitarsasag"

            refer="Key\_legitarsasag">

            <xs:selector xpath="Repülö" />

            <xs:field xpath="@LTársaságID" />

        </xs:keyref>

        <xs:keyref name="FKey\_repulo\_repulo\_tipus"

            refer="Key\_rtipus">

            <xs:selector xpath="Repülö" />

            <xs:field xpath="@RTipusID" />

        </xs:keyref>

        <xs:keyref name="FKey\_jarat\_repulo"

        refer="Key\_repulo">

            <xs:selector xpath="Járat" />

            <xs:field xpath="@RepülöID" />

        </xs:keyref>

        <xs:keyref name="FKey\_utas\_jarat"

        refer="Key\_jarat">

            <xs:selector xpath="Utas" />

            <xs:field xpath="@JáratID" />

        </xs:keyref>

    </xs:element>

    <!--Sajáttípusok meghatározás-->

    <xs:simpleType name="emailTipus">

        <xs:restriction base="xs:string">

            <!--Egyszerü email-->

            <xs:pattern value="[a-zA-Z]{0,30}[0-9]{0,4}[@][a-z]{2,9}[.][a-z]{2,3}" />

        </xs:restriction>

    </xs:simpleType>

    <xs:simpleType name="telefonTipus">

        <xs:restriction base="xs:string">

            <!--+36XX-XXX-XXXX-->

            <xs:pattern value="[+][3]{1}[6]{1}[1-9]{1}[0-9]{1}[-][0-9]{3}[-][0-9]{4}" />

            <!--+361-XXX-XXXX-->

            <xs:pattern value="[+][3]{1}[6]{1}[1]{1}[-][0-9]{3}[-][0-9]{4}"/>

        </xs:restriction>

    </xs:simpleType>

    <!--IATA kód https://en.wikipedia.org/wiki/International\_Air\_Transport\_Association -->

    <xs:simpleType name="indulasCelTipus">

        <xs:restriction base="xs:string">

            <xs:pattern value="[A-Z]{3}" />

        </xs:restriction>

    </xs:simpleType>

    <!--egyszerü idopont 2023-2025.MM.DD-->

    <xs:simpleType name="idopontTipus">

        <xs:restriction base="xs:string">

            <xs:pattern value="[2][0][2][3-5][.][0-1][0-9][.][0-3][0-9]" />

        </xs:restriction>

    </xs:simpleType>

</xs:schema>

**2. Feladat**

A feladat egy DOM program létrehozása volt, ami adminisztrációs feladatok ellátására képes, mint például olvasni, írni, módosítani, keresni a fájlban.

**2a) Adatolvasás:**

Beolvassuk a teljes dokumentumot majd kiíratjuk a konzolra a benne szereplő adatokat strukturált formában.

**package** hu.domparse.yg5yao;

**import** java.io.File;

**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;

**public** **class** DomReadYG5YAO {

**public** **static** **void** main(String[] args) {

NodeList list;

**try** {

DocumentBuilderFactory factory = DocumentBuilderFactory.*newInstance*();

DocumentBuilder builder = factory.newDocumentBuilder();

// fájl beolvasása

Document document = builder.parse(**new** File("../XMLYG5YAO.xml"));

document.getDocumentElement().normalize();

// Gyökér elem megkeresése és kiíratás

System.***out***.println("<" + document.getDocumentElement().getNodeName()+">");

// Aktuális elem meghatározása (Légitársaság)

list = document.getElementsByTagName("Légitársaság");

**for** (**int** i = 0; i < list.getLength(); i++) {

Node node = list.item(i);

// Légitársaság adatainak kiirása

**if** (node.getNodeType() == Node.***ELEMENT\_NODE***) {

Element element = (Element) node;

System.***out***.println("\t<"+node.getNodeName()+" "+"LTársaságID="+ "\""+element.getAttribute("LTársaságID")+ "\""+">" );

System.***out***.println("\t\t<Név>" + element.getElementsByTagName("Név").item(0).getTextContent()+"</Név>");

System.***out***.println("\t\t<Weboldal>" + element.getElementsByTagName("Weboldal").item(0).getTextContent()+"</Weboldal>");

System.***out***.println("\t\t<Elérhetöség>");

// Légitársaság elérhetőségeinek kiirása

Node elerhetoseg = list.item(i);

**if** (elerhetoseg.getNodeType() == Node.***ELEMENT\_NODE***) {

Element elementelerhetoseg = (Element) node;

System.***out***.println("\t\t\t<E-mailcím>" + elementelerhetoseg.getElementsByTagName("E-mailcím").item(0).getTextContent()+"</E-mailcím>");

System.***out***.println("\t\t\t<Telefonszám>" + elementelerhetoseg.getElementsByTagName("Telefonszám").item(0).getTextContent()+"</Telefonszám>");

}

System.***out***.println("\t\t<Elérhetöség>");

System.***out***.println("\t</"+node.getNodeName()+">");

}

}

// Aktuális elem meghatározása (Repülö\_Tipus)

list = document.getElementsByTagName("Repülö\_Tipus");

**for** (**int** i = 0; i < list.getLength(); i++) {

Node node = list.item(i);

// Repülö\_Tipus adatainak kiirása

**if** (node.getNodeType() == Node.***ELEMENT\_NODE***) {

Element element = (Element) node;

System.***out***.println("\t<"+node.getNodeName()+" "+"RTipusID="+ "\""+element.getAttribute("RTipusID")+ "\""+">" );

System.***out***.println("\t\t<Gyartó>" + element.getElementsByTagName("Gyartó").item(0).getTextContent()+"</Gyartó>");

System.***out***.println("\t\t<Elnevezés>" + element.getElementsByTagName("Elnevezés").item(0).getTextContent()+"</Elnevezés>");

System.***out***.println("\t\t<Hajtómű>" + element.getElementsByTagName("Hajtómű").item(0).getTextContent()+"</Hajtómű>");

System.***out***.println("\t</"+node.getNodeName()+">");

}

}

// Aktuális elem meghatározása (Flotta)

list = document.getElementsByTagName("Flotta");

**for** (**int** i = 0; i < list.getLength(); i++) {

Node node = list.item(i);

// Flotta adatainak kiirása

**if** (node.getNodeType() == Node.***ELEMENT\_NODE***) {

Element element = (Element) node;

System.***out***.println("\t<"+node.getNodeName()+" "+"RTipusID="+ "\""+element.getAttribute("RTipusID")+" "+"LTársaságID="+ "\""+element.getAttribute("LTársaságID")+ "\""+">" );

System.***out***.println("\t\t<Darabszám>" + element.getElementsByTagName("Darabszám").item(0).getTextContent()+"</Darabszám>");

System.***out***.println("\t</"+node.getNodeName()+">");

}

}

// Aktuális elem meghatározása (Repülö)

list = document.getElementsByTagName("Repülö");

**for** (**int** i = 0; i < list.getLength(); i++) {

Node node = list.item(i);

// Repülö adatainak kiirása

**if** (node.getNodeType() == Node.***ELEMENT\_NODE***) {

Element element = (Element) node;

System.***out***.println("\t<"+node.getNodeName()+" "+"RepülöID="+ "\""+element.getAttribute("RepülöID")+" "+"RTipusID="+ "\""+element.getAttribute("RTipusID")+" "+"LTársaságID="+ "\""+element.getAttribute("LTársaságID")+ "\""+">" );

System.***out***.println("\t\t<Gyártásiszám>" + element.getElementsByTagName("Gyártásiszám").item(0).getTextContent()+"</Gyártásiszám>");

System.***out***.println("\t\t<Hívószám>" + element.getElementsByTagName("Hívószám").item(0).getTextContent()+"</Hívószám>");

//Személyzetek lekérése

NodeList links = element.getElementsByTagName("Személyzet");

**for**(**int** j=0; j<links.getLength(); j++)

{

Node link = links.item(j);

System.***out***.println("\t\t<Személyzet>" +link.getTextContent()+"</Személyzet>");

}

System.***out***.println("\t</"+node.getNodeName()+">");

}

}

// Aktuális elem meghatározása (Járat)

list = document.getElementsByTagName("Járat");

**for** (**int** i = 0; i < list.getLength(); i++) {

Node node = list.item(i);

// Járat adatainak kiirása

**if** (node.getNodeType() == Node.***ELEMENT\_NODE***) {

Element element = (Element) node;

System.***out***.println("\t<"+node.getNodeName()+" "+"JáratID="+ "\""+element.getAttribute("JáratID")+" "+"RepülöID="+ "\""+element.getAttribute("RepülöID")+ "\""+">" );

System.***out***.println("\t\t<Indulás>" + element.getElementsByTagName("Indulás").item(0).getTextContent()+"</Indulás>");

System.***out***.println("\t\t<Cél>" + element.getElementsByTagName("Cél").item(0).getTextContent()+"</Cél>");

System.***out***.println("\t\t<Idöpont>" + element.getElementsByTagName("Idöpont").item(0).getTextContent()+"</Idöpont>");

System.***out***.println("\t</"+node.getNodeName()+">");

}

}

// Aktuális elem meghatározása (Utas)

list = document.getElementsByTagName("Utas");

**for** (**int** i = 0; i < list.getLength(); i++) {

Node node = list.item(i);

// Utas adatainak kiirása

**if** (node.getNodeType() == Node.***ELEMENT\_NODE***) {

Element element = (Element) node;

System.***out***.println("\t<"+node.getNodeName()+" "+"UtasID="+ "\""+element.getAttribute("UtasID")+" "+"JáratID="+ "\""+element.getAttribute("JáratID")+ "\""+">" );

System.***out***.println("\t\t<Név>" + element.getElementsByTagName("Név").item(0).getTextContent()+"</Név>");

System.***out***.println("\t\t<Elérhetöség>");

// Utas elérhetőségeinek kiirása

Node elerhetoseg = list.item(i);

**if** (elerhetoseg.getNodeType() == Node.***ELEMENT\_NODE***) {

Element elementelerhetoseg = (Element) node;

System.***out***.println("\t\t\t<E-mailcím>" + elementelerhetoseg.getElementsByTagName("E-mailcím").item(0).getTextContent()+"</E-mailcím>");

System.***out***.println("\t\t\t<Telefonszám>" + elementelerhetoseg.getElementsByTagName("Telefonszám").item(0).getTextContent()+"</Telefonszám>");

}

System.***out***.println("\t\t<Elérhetöség>"); System.***out***.println("\t\t<Poggyász>" + element.getElementsByTagName("Poggyász").item(0).getTextContent()+"</Poggyász>");

System.***out***.println("\t</"+node.getNodeName()+">");

}

}

System.***out***.println("</" + document.getDocumentElement().getNodeName()+">");

} **catch** (ParserConfigurationException e) {

e.printStackTrace();

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

**2b) Adatmódosítás:**

Megkeresi a módosítandó elemet ID segítségével, majd megnézi mit kell módosítani rajta és mire. Ezért queryIDChangeThings felel. Kiíratásért egy másik java fájlban található funkció felel.

**package** hu.domparse.yg5yao;

**import** java.io.File;

**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** org.w3c.dom.Document;

**import** org.w3c.dom.Element;

**import** org.w3c.dom.Node;

**import** org.w3c.dom.NodeList;

**public** **class** DomModifyYG5YAO {

**public** **static** **void** main(String[] args) {

**try** {

DocumentBuilderFactory factory = DocumentBuilderFactory.*newInstance*();

DocumentBuilder builder = factory.newDocumentBuilder();

// fájl beolvasása

Document document = builder.parse(**new** File("../XMLYG5YAO.xml"));

document.getDocumentElement().normalize();

// Gyökér elem megkeresése

System.***out***.println("<" + document.getDocumentElement().getNodeName()+">");

//utas adatmódosítások(2db)

System.***out***.println("<!-------Modosítás előtt------->");

hu.domparse.yg5yao.DomQueryYG5YAO.*queryID*(document, "Utas", "003");

*queryIDChangeThings*(document, "Utas", "003","Poggyász","1db nagy");

System.***out***.println("<!-------Modosítás után------->");

hu.domparse.yg5yao.DomQueryYG5YAO.*queryID*(document, "Utas", "003");

System.***out***.println("<!-------Modosítás előtt------->");

hu.domparse.yg5yao.DomQueryYG5YAO.*queryID*(document, "Utas", "003");

*queryIDChangeThings*(document, "Utas", "003","E-mailcím","smklevente@gmail.com");

System.***out***.println("<!-------Modosítás után------->");

hu.domparse.yg5yao.DomQueryYG5YAO.*queryID*(document, "Utas", "003");

//repülö módosítás (1db)

System.***out***.println("<!-------Modosítás előtt------->");

hu.domparse.yg5yao.DomQueryYG5YAO.*queryID*(document, "Repülö", "AA\_01");

*queryIDChangeThings*(document, "Repülö", "AA\_01","Hívószám","AA\_02");

System.***out***.println("<!-------Modosítás után------->");

hu.domparse.yg5yao.DomQueryYG5YAO.*queryID*(document, "Repülö", "AA\_01");

//Légitársaségok módosítása (2db)

System.***out***.println("<!-------Modosítás előtt------->");

hu.domparse.yg5yao.DomQueryYG5YAO.*queryID*(document, "Légitársaság", "AA");

*queryIDChangeThings*(document, "Légitársaság", "AA","E-mailcím","customerSupport@aa.com");

System.***out***.println("<!-------Modosítás után------->");

hu.domparse.yg5yao.DomQueryYG5YAO.*queryID*(document, "Légitársaság", "AA");

System.***out***.println("<!-------Modosítás előtt------->");

hu.domparse.yg5yao.DomQueryYG5YAO.*queryID*(document, "Légitársaság", "AA");

*queryIDChangeThings*(document, "Légitársaság", "AA","Név","American Airlines");

System.***out***.println("<!-------Modosítás után------->");

hu.domparse.yg5yao.DomQueryYG5YAO.*queryID*(document, "Légitársaság", "AA");

System.***out***.println("</" + document.getDocumentElement().getNodeName()+">");

Transformer transformer = TransformerFactory.*newInstance*().newTransformer();

Source input = **new** DOMSource(document);

Result output = **new** StreamResult(**new** File("../XMLYG5YAOModify.xml"));

transformer.transform(input, output);

} **catch** (ParserConfigurationException e) {

e.printStackTrace();

} **catch** (Exception e) {

e.printStackTrace();

}

}

**public** **static** **void** queryIDChangeThings(Document document, String elemKeres, String mitKeres, String holValtoztat, String mireValtoztat) {

String holKeres;

**if**(elemKeres=="Légitársaság") {

holKeres="LTársaságID";

}

**else** **if**(elemKeres=="Repülö\_Tipus")

{

holKeres="RTipusID";

}

**else** {

holKeres= elemKeres+"ID";

}

NodeList list = document.getElementsByTagName(elemKeres);

**for** (**int** i = 0; i < list.getLength(); i++) {

Node node = list.item(i);

**if** (node.getNodeType() == Node.***ELEMENT\_NODE***) {

Element element = (Element) node;

Node nodecim = list.item(i);

**if** (nodecim.getNodeType() == Node.***ELEMENT\_NODE***) {

Element elemValtoztat = (Element) node;

// lekéri az id-t és megnézi egyezik-e a keresettel

**if** (element.getAttribute(holKeres).equals(mitKeres)) {

elemValtoztat.getElementsByTagName(holValtoztat).item(0).setTextContent(mireValtoztat);

}

}

}

}

}

}

**2c) Adat lekérdezés:**

Itt két féle képen lehet keresni adatra. Első, ahol az XML dokumentumban megkeresi az összes megadott egyedet és kiírja a keresett adatot. Második, megkeresi a dokumentumban az egyed ID alapján és kiírja az összes hozzá tartozó adatot.

**package** hu.domparse.yg5yao;

**import** java.io.File;

**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;

**public** **class** DomQueryYG5YAO {

**public** **static** **void** main(String[] args) {

**try** {

DocumentBuilderFactory factory = DocumentBuilderFactory.*newInstance*();

DocumentBuilder builder = factory.newDocumentBuilder();

// fájl beolvasása

Document document = builder.parse(**new** File("../XMLYG5YAO.xml"));

document.getDocumentElement().normalize();

// Gyökér elem megkeresése

System.***out***.println("<" + document.getDocumentElement().getNodeName()+">");

// Repülök gyártási számai

*query*(document, "Repülö", "Gyártásiszám");

System.***out***.println("<---------------->");

// Légitársaság nevei

*query*(document, "Légitársaság", "Név");

System.***out***.println("<---------------->");

//Utas adatai kiírása id alapján

*queryID*(document, "Utas", "012");

System.***out***.println("<---------------->");

//Légitársaság adatai kiírása id alapján

*queryID*(document, "Légitársaság", "LH");

System.***out***.println("<---------------->");

//Repülö adatainak kiírása id alapján

*queryID*(document, "Repülö", "LH\_02");

System.***out***.println("<---------------->");

//Repülö adatainak kiírása id alapján

*queryID*(document, "Járat", "J\_03");

System.***out***.println("</" + document.getDocumentElement().getNodeName()+">");

} **catch** (ParserConfigurationException e) {

e.printStackTrace();

} **catch** (Exception e) {

e.printStackTrace();

}

}

**public** **static** **void** query(Document document, String elemKeres, String mitKeres) {

// Aktuális elem meghatározása

NodeList list = document.getElementsByTagName(elemKeres);

**for** (**int** i = 0; i < list.getLength(); i++) {

NodeList nodeList = list.item(i).getChildNodes();

System.***out***.println("<"+list.item(i).getNodeName()+">");

**for** (**int** k = 0; k < nodeList.getLength(); k++) {

Node gye = nodeList.item(k);

**if** (gye.getNodeName().compareTo(mitKeres) == 0) {

System.***out***.println("\t<"+gye.getNodeName()+">"+ gye.getTextContent()+"</"+gye.getNodeName()+">");

}

}

System.***out***.println("</"+list.item(i).getNodeName()+">");

}

}

**public** **static** **void** queryID(Document document, String elemKeres, String mitKeres) {

// Aktuális elem meghatározása

String holKeres;

**if**(elemKeres=="Légitársaság") {

holKeres="LTársaságID";

}

**else** **if**(elemKeres=="Repülö\_Tipus")

{

holKeres="RTipusID";

}

**else** {

holKeres= elemKeres+"ID";

}

NodeList list = document.getElementsByTagName(elemKeres);

**for** (**int** i = 0; i < list.getLength(); i++) {

Node node = list.item(i);

Element element = (Element) node;

// Adatok kiirása

**if** (node.getNodeType() == Node.***ELEMENT\_NODE*** && element.getAttribute(holKeres).equals(mitKeres)) {

NodeList nodeList = list.item(i).getChildNodes();

System.***out***.println("<"+list.item(i).getNodeName()+" "+holKeres+"= "+"\""+element.getAttribute(holKeres)+"\""+">");

**for** (**int** k = 0; k < nodeList.getLength(); k++) {

Node gye = nodeList.item(k);

**if** (!gye.getNodeName().equals("#text")) {

**if**(!gye.getNodeName().equals("Elérhetöség")) {

System.***out***.println("\t<"+gye.getNodeName()+">"+ gye.getTextContent()+"</"+gye.getNodeName()+">");

}

**else** {

System.***out***.println("\t<"+gye.getNodeName()+">");

NodeList gyeList =gye.getChildNodes();

**for**(**int** l=0;l<gyeList.getLength();l++) {

Node gyeNode = gyeList.item(l);

**if** (!gyeNode.getNodeName().equals("#text")) {

System.***out***.println("\t\t<"+gyeNode.getNodeName()+">"+ gyeNode.getTextContent()+"</"+gyeNode.getNodeName()+">");

}

}

System.***out***.println("\t</"+gye.getNodeName()+">");

}

}

}

System.***out***.println("</"+list.item(i).getNodeName()+">");

}

}

}

}

**2d) Adatírás:**

A programunk megírása közben feltételezzük, hogy ismerjük az eredeti dokumentum struktúráját, és azt felhasználva hozzuk létre az elejétől az új XML-t. Hasonló módon alapszik, mint az adat olvasás csak itt kiírás helyett belerakjuk az adatot az új dokumentumba és azt írjuk ki utána mind konzolra mind a fájlba.

**package** hu.domparse.yg5yao;

**import** java.io.FileOutputStream;

**import** java.io.File;

**import** java.io.FileNotFoundException;

**import** java.io.IOException;

**import** java.io.OutputStream;

**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** org.w3c.dom.Document;

**import** org.w3c.dom.Element;

**import** org.w3c.dom.NamedNodeMap;

**import** org.w3c.dom.Node;

**import** org.w3c.dom.NodeList;

**public** **class** DomWriteYG5YAO {

**public** **static** **void** main(String[] args) **throws** ParserConfigurationException, TransformerException, FileNotFoundException{

NodeList list;

DocumentBuilderFactory docFactory = DocumentBuilderFactory.*newInstance*();

DocumentBuilder docBuilder = docFactory.newDocumentBuilder();

FileOutputStream file = **new** FileOutputStream("../XMLYG5YAO1.xml");

**try** {

Document base = docBuilder.parse(**new** File("../XMLYG5YAO.xml"));

base.getDocumentElement().normalize();

//RootElement

Document doc= docBuilder.newDocument();

String baseString=base.getDocumentElement().getNodeName();

Element rootElement = doc.createElement(baseString);

doc.appendChild(rootElement);

list=base.getElementsByTagName("Légitársaság");

**for**(**int** i=0; i<list.getLength();i++) {

Node node = list.item(i);

//Légitársaság element letrehozas

Element element = (Element) node;

Element Ltarsasag = doc.createElement(element.getNodeName());

**if**(node.getNodeType()==Node.***ELEMENT\_NODE***) {

rootElement.appendChild(Ltarsasag);

//checks for attributes then adds them

**if**(node.hasAttributes()) {

NamedNodeMap nodeMap = node.getAttributes();

**for**(**int** j=0;j<nodeMap.getLength();j++) {

Node atrNode = nodeMap.item(j);

Ltarsasag.setAttribute(atrNode.getNodeName(), atrNode.getNodeValue());

}

}

}

//név element

Element Ltnev = doc.createElement("Név");

Ltnev.setTextContent(element.getElementsByTagName("Név").item(0).getTextContent());

Ltarsasag.appendChild(Ltnev);

//weboldal element

Element webold = doc.createElement("Weboldal");

webold.setTextContent(element.getElementsByTagName("Weboldal").item(0).getTextContent());

Ltarsasag.appendChild(webold);

Node elerhetosegNode = list.item(i);

//elerhetoseg

**if**(elerhetosegNode.getNodeType()==Node.***ELEMENT\_NODE***) {

Element elerhetoBase = (Element) node;

Element elerhetoNew = doc.createElement("Elérhetöség");

Ltarsasag.appendChild(elerhetoNew);

//email element

Element nap = doc.createElement("E-mailcím");

nap.setTextContent(elerhetoBase.getElementsByTagName("E-mailcím").item(0).getTextContent());

elerhetoNew.appendChild(nap);

//telefonszam element

Element kezdet = doc.createElement("Telefonszám");

kezdet.setTextContent(elerhetoBase.getElementsByTagName("Telefonszám").item(0).getTextContent());

elerhetoNew.appendChild(kezdet);

}

}

list=base.getElementsByTagName("Repülö\_Tipus");

**for**(**int** i=0; i<list.getLength();i++) {

Node node = list.item(i);

//Repülö\_Tipus element

Element element = (Element) node;

Element rTipus = doc.createElement(element.getNodeName());

**if**(node.getNodeType()==Node.***ELEMENT\_NODE***) {

rootElement.appendChild(rTipus);

**if**(node.hasAttributes()) {

NamedNodeMap nodeMap = node.getAttributes();

**for**(**int** j=0;j<nodeMap.getLength();j++) {

Node atrNode = nodeMap.item(j);

rTipus.setAttribute(atrNode.getNodeName(), atrNode.getNodeValue());

}

}

}

//gyarto element

Element gyarto = doc.createElement("Gyartó");

gyarto.setTextContent(element.getElementsByTagName("Gyartó").item(0).getTextContent());

rTipus.appendChild(gyarto);

//elnevezes element

Element elnev = doc.createElement("Elnevezés");

elnev.setTextContent(element.getElementsByTagName("Elnevezés").item(0).getTextContent());

rTipus.appendChild(elnev);

//hajtomu element

Element hajtmu = doc.createElement("Hajtómű");

hajtmu.setTextContent(element.getElementsByTagName("Hajtómű").item(0).getTextContent());

rTipus.appendChild(hajtmu);

}

list=base.getElementsByTagName("Flotta");

**for**(**int** i=0; i<list.getLength();i++) {

Node node = list.item(i);

//flotta element

Element element = (Element) node;

Element flotta = doc.createElement(element.getNodeName());

**if**(node.getNodeType()==Node.***ELEMENT\_NODE***) {

rootElement.appendChild(flotta);

**if**(node.hasAttributes()) {

NamedNodeMap nodeMap = node.getAttributes();

**for**(**int** j=0;j<nodeMap.getLength();j++) {

Node atrNode = nodeMap.item(j);

flotta.setAttribute(atrNode.getNodeName(), atrNode.getNodeValue());

}

}

}

//darabszam element

Element db = doc.createElement("Darabszám");

db.setTextContent(element.getElementsByTagName("Darabszám").item(0).getTextContent());

flotta.appendChild(db);

}

list=base.getElementsByTagName("Repülö");

**for**(**int** i=0; i<list.getLength();i++) {

Node node = list.item(i);

//repulo element

Element element = (Element) node;

Element repulo = doc.createElement(element.getNodeName());

**if**(node.getNodeType()==Node.***ELEMENT\_NODE***) {

rootElement.appendChild(repulo);

**if**(node.hasAttributes()) {

NamedNodeMap nodeMap = node.getAttributes();

**for**(**int** j=0;j<nodeMap.getLength();j++) {

Node atrNode = nodeMap.item(j);

repulo.setAttribute(atrNode.getNodeName(), atrNode.getNodeValue());

}

}

}

//gyartasiszam element

Element factoryNumber = doc.createElement("Gyártásiszám");

factoryNumber.setTextContent(element.getElementsByTagName("Gyártásiszám").item(0).getTextContent());

repulo.appendChild(factoryNumber);

//hivojel element

Element callsign = doc.createElement("Hívószám");

callsign.setTextContent(element.getElementsByTagName("Hívószám").item(0).getTextContent());

repulo.appendChild(callsign);

//szemelyzet element

NodeList links = element.getElementsByTagName("Személyzet");

**for**(**int** j=0; j<links.getLength(); j++)

{

Node link = links.item(j);

Element szemelyzet = doc.createElement("Személyzet");

szemelyzet.setTextContent(link.getTextContent());

repulo.appendChild(szemelyzet);

}

}

list=base.getElementsByTagName("Járat");

**for**(**int** i=0; i<list.getLength();i++) {

Node node = list.item(i);

//jarat element

Element element = (Element) node;

Element jarat = doc.createElement(element.getNodeName());

**if**(node.getNodeType()==Node.***ELEMENT\_NODE***) {

rootElement.appendChild(jarat);

**if**(node.hasAttributes()) {

NamedNodeMap nodeMap = node.getAttributes();

**for**(**int** j=0;j<nodeMap.getLength();j++) {

Node atrNode = nodeMap.item(j);

jarat.setAttribute(atrNode.getNodeName(), atrNode.getNodeValue());

}

}

}

//indulas element

Element from = doc.createElement("Indulás");

from.setTextContent(element.getElementsByTagName("Indulás").item(0).getTextContent());

jarat.appendChild(from);

//cel element

Element to = doc.createElement("Cél");

to.setTextContent(element.getElementsByTagName("Cél").item(0).getTextContent());

jarat.appendChild(to);

//idopont element

Element date = doc.createElement("Idöpont");

date.setTextContent(element.getElementsByTagName("Idöpont").item(0).getTextContent());

jarat.appendChild(date);

}

list=base.getElementsByTagName("Utas");

**for**(**int** i=0; i<list.getLength();i++) {

Node node = list.item(i);

//utas element letrehozas

Element element = (Element) node;

Element utas = doc.createElement(element.getNodeName());

**if**(node.getNodeType()==Node.***ELEMENT\_NODE***) {

rootElement.appendChild(utas);

//checks for attributes then adds them

**if**(node.hasAttributes()) {

NamedNodeMap nodeMap = node.getAttributes();

**for**(**int** j=0;j<nodeMap.getLength();j++) {

Node atrNode = nodeMap.item(j);

utas.setAttribute(atrNode.getNodeName(), atrNode.getNodeValue());

}

}

}

//név element

Element uNev = doc.createElement("Név");

uNev.setTextContent(element.getElementsByTagName("Név").item(0).getTextContent());

utas.appendChild(uNev);

Node elerhetosegNode = list.item(i);

//elerhetoseg

**if**(elerhetosegNode.getNodeType()==Node.***ELEMENT\_NODE***) {

Element elerhetoBase = (Element) node;

Element elerhetoNew = doc.createElement("Elérhetöség");

utas.appendChild(elerhetoNew);

//email element

Element nap = doc.createElement("E-mailcím");

nap.setTextContent(elerhetoBase.getElementsByTagName("E-mailcím").item(0).getTextContent());

elerhetoNew.appendChild(nap);

//telefon element

Element kezdet = doc.createElement("Telefonszám");

kezdet.setTextContent(elerhetoBase.getElementsByTagName("Telefonszám").item(0).getTextContent());

elerhetoNew.appendChild(kezdet);

}

//poggyasz element

Element bagge = doc.createElement("Poggyász");

bagge.setTextContent(element.getElementsByTagName("Poggyász").item(0).getTextContent());

utas.appendChild(bagge);

}

*writeXML*(doc, System.***out***, file);

}

**catch**(Exception e) {

e.printStackTrace();

}

}

//Kiiras file, console

**private** **static** **void** writeXML(Document doc, OutputStream output, FileOutputStream file) **throws** TransformerException {

TransformerFactory transFact = TransformerFactory.*newInstance*();

Transformer trans = transFact.newTransformer();

trans.setOutputProperty(OutputKeys.***INDENT***, "yes");

DOMSource source = **new** DOMSource(doc);

StreamResult result = **new** StreamResult(output);

StreamResult resultFile = **new** StreamResult(file);

trans.transform(source, result);

trans.transform(source, resultFile);

}

}