

# JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Repülőgép járat nyilvántartó adatbázis

Készítette: **Simkó Levente**

Neptunkód: **YG5YAO**

Dátum: **2023. 12. 10.**

## Tartalomjegyzék

<b>A Feladat leírása:</b> .....	<b>2</b>
<b>1.Feladat</b> .....	<b>3</b>
1a) Az adatbázis ER modellje .....	3
1b) Konvertálás XDM modellre .....	4
1c) XML az XDM alapján .....	5
1d) XMLSchema XML alapján .....	7
<b>2.Feladat</b> .....	<b>11</b>
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 járatához kapcsolódik stb. De vegyük például a légitársaságot, 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.)

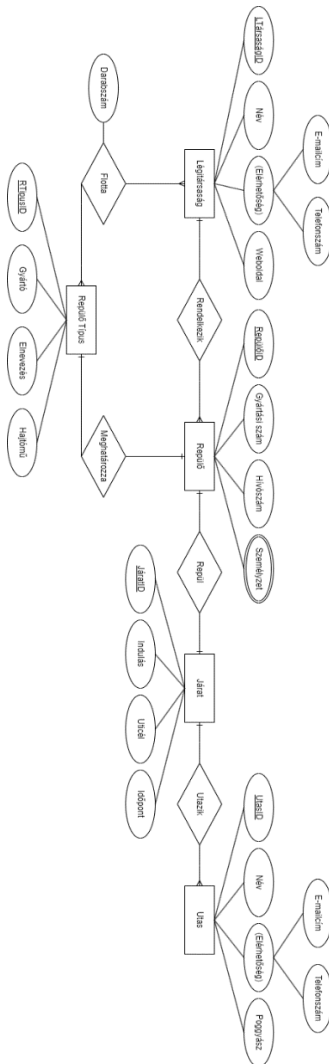
Utastak nem sokban különbözik az eltárolt adata, mint a légitárságnak, csak weboldal helyett a poggyászt tároljuk el, hogy mekkora és mennyi, valamint a járat ID-ját, amin utazik. ID-nek csak sima 3 számból készült kombinációt adtam meg.

Az általam fentebb felsorolt adatokat vettem fel az adatbázisba.

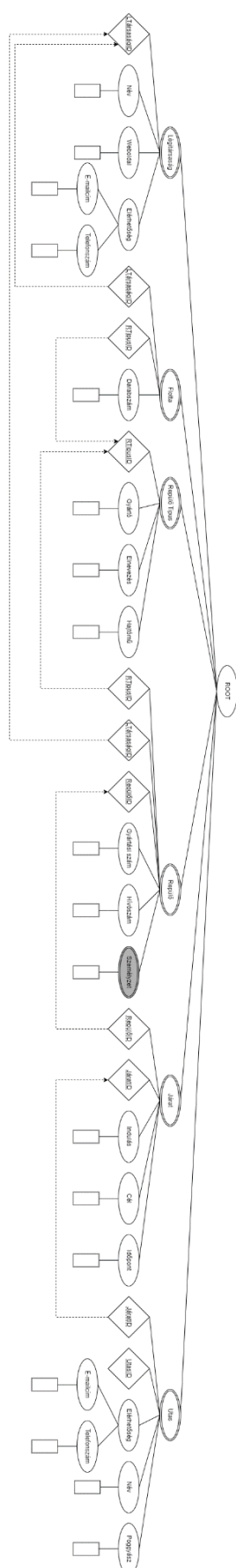
## 1.Feladat

### 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 járhoz lehet rendelve így ott 1:1 kapcsolat van. Járaton viszont több utas utazik, így ott 1:N kapcsolat van.



## 1b) Konvertálás XDM modellre:



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="XMLSchemaYG5YA0.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>
    <Elérhetőség>
      <E-mailcím>contactCustomer@lh.com</E-mailcím>
      <Telefonszám>+361-429-2200</Telefonszám>
    </Elérhetőség>
  </Légitársaság>
  <!--Lt British Airways egyed-->
  <Légitársaság LTársaságID="BA">
    <Név>British Airways</Név>
    <Weboldal>britishairways.com</Weboldal>
    <Elérhetőség>
      <E-mailcím>customerContact@ba.com</E-mailcím>
      <Telefonszám>+361-777-4747</Telefonszám>
    </Elérhetőség>
  </Légitársaság>
  <!--Lt American Airlines egyed-->
  <Légitársaság LTársaságID="AA">
    <Név>American Airlines, Inc.</Név>
    <Weboldal>www.aa.com</Weboldal>
    <Elérhetőség>
      <E-mailcím>customerHelp@aa.com</E-mailcím>
      <Telefonszám>+361-429-2346</Telefonszám>
    </Elérhetőség>
  </Légitársaság>
  <!--RT Airbus 380 egyed-->
  <Repülő_Típus RTípusID="A380">
    <Gyártó>Airbus</Gyártó>
    <Elnevezés>Airbus A380</Elnevezés>
    <Hajtómű>4x Rolls-Royce Trent 900</Hajtómű>
  </Repülő_Típus>
  <!--RT Boeing 777 egyed-->
  <Repülő_Típus RTípusID="B777">
    <Gyártó>Boeing Commercial Airplanes</Gyártó>
    <Elnevezés>Boeing 777</Elnevezés>
    <Hajtómű>2x General Electric GE90</Hajtómű>
```

```
</Repülő_Típus>
<!--RT Boeing 787 Dreamliner-->
<Repülő_Típus RTípusID="B787D">
  <Gyártó>Boeing Commercial Airplanes</Gyártó>
  <Elnevezés>Boeing 787 Dreamliner</Elnevezés>
  <Hajtómű>2x Rolls-Royce Trent 1000</Hajtómű>
</Repülő_Típus>
<!--F LH B787D egyed-->
<Flotta RTípusID="B787D" LTársaságID="LH">
  <Darabszám>358</Darabszám>
</Flotta>
<!--F AA A380 egyed-->
<Flotta RTípusID="A380" LTársaságID="AA">
  <Darabszám>102</Darabszám>
</Flotta>
<!--F BA B777 egyed-->
<Flotta RTípusID="B777" LTársaságID="BA">
  <Darabszám>246</Darabszám>
</Flotta>
<!--Repülő egyedek-->
<Repülő RepülőID="AA_01" RTípusID="A380" LTársaságID="AA">
  <Gyártásiszám>A380-122-449</Gyártásiszám>
  <Hívószám>AA-01</Hívószám>
  <Személyzet>Piloták</Személyzet>
  <Személyzet>Légiutaskisérők</Személyzet>
</Repülő>
<Repülő RepülőID="LH_02" RTípusID="B787D" LTársaságID="LH">
  <Gyártásiszám>B787-122-001</Gyártásiszám>
  <Hívószám>LH-02</Hívószám>
  <Személyzet>Piloták</Személyzet>
  <Személyzet>Légiutaskisérők</Személyzet>
</Repülő>
<Repülő RepülőID="BA_03" RTípusID="B777" LTársaságID="BA">
  <Gyártásiszám>B777-287-019</Gyártásiszám>
  <Hívószám>BA-03</Hívószám>
  <Személyzet>Piloták</Személyzet>
  <Személyzet>Légiutaskisérők</Személyzet>
</Repülő>
<!--Járatok-->
<Járat JáratID="J_01" RepülőID="BA_03">
  <Indulás>LHR</Indulás>
  <Cél>BUD</Cél>
  <Időpont>2024.01.12</Időpont>
</Járat>
<Járat JáratID="J_02" RepülőID="LH_02">
  <Indulás>BER</Indulás>
  <Cél>BUD</Cél>
  <Időpont>2024.02.09</Időpont>
</Járat>
```

```

<Járat JáratID="J_03" RepülőID="AA_01">
  <Indulás>LAX</Indulás>
  <Cél>LHR</Cél>
  <Időpont>2024.01.24</Időpont>
</Járat>
<!--Utasok-->
<Utas UtasID="003" JáratID="J_01">
  <Név>Simkó Levente</Név>
  <Elérhetőség>
    <E-mailcím>smlevente@gmail.com</E-mailcím>
    <Telefonszám>+3670-429-5555</Telefonszám>
  </Elérhetőség>
  <Poggyász>2db kézi</Poggyász>
</Utas>
<Utas UtasID="012" JáratID="J_03">
  <Név>Magyar Márk</Név>
  <Elérhetőség>
    <E-mailcím>mmark1996@outlook.com</E-mailcím>
    <Telefonszám>+3630-479-8888</Telefonszám>
  </Elérhetőség>
  <Poggyász>1db nagy</Poggyász>
</Utas>
<Utas UtasID="048" JáratID="J_02">
  <Név>Kiss Szabina</Név>
  <Elérhetőség>
    <E-mailcím>szabinakiss14@gmail.com</E-mailcím>
    <Telefonszám>+3620-888-2200</Telefonszám>
  </Elérhetőség>
  <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/XMLSchemaYG5YA0"
  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="Elérhetőség">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="E-
mailcím" type="emailTipus" />
                    <xs:element
name="Telefonszám" type="telefonTipus" />
                </xs:sequence>
            </xs:complexType>
        </xs:element>
        <xs:attribute name="LTársaságID" type="xs:string" />
    </xs:complexType>
</xs:element>
<!--Repülő_Típus meghatározás-->
<xs:element name="Repülő_Típus" minOccurs="1"
maxOccurs="unbounded">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Gyártó" type="xs:string"/>
            <xs:element name="Elnevezés" type="xs:string"/>
            <xs:element name="Hajtómű" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="RTípusID" 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="RTípusID" type="xs:string" />
    </xs:complexType>
</xs:element>
<!--Repülő meghatározás-->
<xs:element name="Repülő" 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="RepülőID" type="xs:string" />
        <xs:attribute name="LTársaságID" type="xs:string" />
        <xs:attribute name="RTípusID" 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="Időpont" type="idopontTipus"/>
                </xs:sequence>
                <xs:attribute name="JáratID" type="xs:string" />
                <xs:attribute name="RepülőID" 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="Elérhetőség">
                        <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="Repülő_Tipus" />
    <xs:field xpath="@RTipusID" />
</xs:key>
<xs:key name="Key_repulo">
    <xs:selector xpath="Repülő" />
    <xs:field xpath="@RepülőID" />

```

```

</xs:key>
<xs:key name="Key_jarat">
  <xs:selector xpath="Jarat" />
  <xs:field xpath="@JaratID" />
</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="Repülő" />
  <xs:field xpath="@LTársaságID" />
</xs:keyref>
<xs:keyref name="FKey_repulo_repulo_tipus"
  refer="Key_rtipus">
  <xs:selector xpath="Repülő" />
  <xs:field xpath="@RTipusID" />
</xs:keyref>
<xs:keyref name="FKey_jarat_repulo"
  refer="Key_repulo">
  <xs:selector xpath="Jarat" />
  <xs:field xpath="@RepülőID" />
</xs:keyref>
<xs:keyref name="FKey_utas_jarat"
  refer="Key_jarat">
  <xs:selector xpath="Utas" />
  <xs:field xpath="@JaratID" />
</xs:keyref>
</xs:element>
<!--Sajáttípusok meghatározás-->
<xs:simpleType name="emailTipus">
  <xs:restriction base="xs:string">
    <!--Egyszerű 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}[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>
<!--egyszerű 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 kiírása
            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<Elérhetőség>");

                // Légitársaság elérhetőségeinek kiírása
                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<Elérhetőség>");
                System.out.println("\t</"+node.getNodeName()+">");
            }
        }
        // Aktuális elem meghatározása (Repülő_Típus)

        list = document.getElementsByTagName("Repülő_Típus");
        for (int i = 0; i < list.getLength(); i++) {
            Node node = list.item(i);
            // Repülő_Típus adatainak kiírása
            if (node.getNodeType() == Node.ELEMENT_NODE) {
                Element element = (Element) node;
                System.out.println("\t<" + node.getNodeName() +
"+"RTípusID="+ "\"" + element.getAttribute("RTípusID") + "\">");
                System.out.println("\t\t<Gyártó>" +
element.getElementsByTagName("Gyártó").item(0).getTextContent()+"</Gyártó>");
                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 kiírása
        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 (Repülő)

    list = document.getElementsByTagName("Repülő");
    for (int i = 0; i < list.getLength(); i++) {
        Node node = list.item(i);
        // Repülő adatainak kiírása
        if (node.getNodeType() == Node.ELEMENT_NODE) {
            Element element = (Element) node;
            System.out.println("\t<" + node.getNodeName() + "
+\"RepülőID=\" + \"\""+element.getAttribute("RepülőID")+\" \"RTipusID=\"+
+\"\""+element.getAttribute("RTipusID")+\" \"LTársaságID=\"+
+\"\""+element.getAttribute("LTársaságID")+ \"\">");
            System.out.println("\t\t<Gyártásiszám>\" +
element.getElementsByTagName("Gyártásiszám").item(0).getTextContent()+"</Gyártási
szám>");
            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 kiírása
        if (node.getNodeType() == Node.ELEMENT_NODE) {
            Element element = (Element) node;
            System.out.println("\t<" + node.getNodeName() + "
+\"JáratID=\" + \"\""+element.getAttribute("JáratID")+\" \"RepülőID=\"+
+\"\""+element.getAttribute("RepülőID")+ \"\">");

```

```

        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<Időpont>" +
element.getElementsByTagName("Időpont").item(0).getTextContent()+"</Időpont>");
        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 kiírása
    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<Elérhetőség>");
        // Utas elérhetőségeinek kiírása
        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<Elérhetőség>");
        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("<-----Módosítás előtt----->");
            hu.domparse.yg5yao.DomQueryYG5YAO.queryID(document, "Utas",
"003");
            queryIDChangeThings(document, "Utas", "003", "Poggyász", "1db
nagy");
            System.out.println("<-----Módosítás után----->");
            hu.domparse.yg5yao.DomQueryYG5YAO.queryID(document, "Utas",
"003");
            System.out.println("<-----Módosítás előtt----->");
            hu.domparse.yg5yao.DomQueryYG5YAO.queryID(document, "Utas",
"003");
            queryIDChangeThings(document, "Utas", "003", "E-
mailcím", "smklevente@gmail.com");
            System.out.println("<-----Módosítás után----->");
            hu.domparse.yg5yao.DomQueryYG5YAO.queryID(document, "Utas",
"003");
            //repülő módosítás (1db)
            System.out.println("<-----Módosítás előtt----->");
            hu.domparse.yg5yao.DomQueryYG5YAO.queryID(document, "Repülő",
"AA_01");
            queryIDChangeThings(document, "Repülő",
"AA_01", "Hívószám", "AA_02");
            System.out.println("<-----Módosítás után----->");
            hu.domparse.yg5yao.DomQueryYG5YAO.queryID(document, "Repülő",
"AA_01");

```

```

//Légitársaságok 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=="Repülő_Típus")
{
holKeres="RTípusID";
}
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](#)



### 2c) Adat lekérdezés:

```
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()+">");

            // Repülők gyártási számai
            query(document, "Repülő", "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("<----->");
        //Repülő adatainak kiírása id alapján
        queryID(document, "Repülő", "LH_02");
        System.out.println("<----->");
        //Repülő 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 == "Repülő_Típus")
        {
            holKeres = "RTípusID";
        }
        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 kiírása
        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("Elérhetőség")) {

                        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 létrehozás
                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).getTe
xtContent());

        Ltarsasag.appendChild(webold);
        Node elerhetosegNode = list.item(i);
        //elerhetoseg
        if(elerhetosegNode.getNodeType()==Node.ELEMENT_NODE) {
            Element elerhetoBase = (Element) node;
            Element elerhetoNew =
doc.createElement("Elérhetőség");
            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("Repülő_Tipus");
    for(int i=0; i<list.getLength();i++) {
        Node node = list.item(i);

        //Repülő_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("Gyártó");

        gyarto.setTextContent(element.getElementsByTagName("Gyártó").item(0).getText
Content());

        rTipus.appendChild(gyarto);
        //elnevezes element
        Element elnev = doc.createElement("Elnevezés");

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

        rTipus.appendChild(elnev);
        //hajtomu element

```

```

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

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

        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).getTextC
ontent());

            flotta.appendChild(db);
        }
        list=base.getElementsByTagName("Repülő");
        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("Időpont");

            date.setTextContent(element.getElementsByTagName("Időpont").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 letrehozás
        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("Elérhetőség");
                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).getTex
tContent());
            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);
    }
}

```