

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Hadsereg struktúra szemléltetése XML
környezetben

Készítette: Urbán Olivér

Neptunkód: HEPMIU

Dátum: 2023.12.05

Tartalomjegyzék

Bevezetés 2.oldal

1.feladat a) ER modell 2.oldal – 3.oldal

b) XDM modell 3.oldal – 4.oldal

c) XML dokumentum 4.oldal – 6.oldal

d) XML Séma dokumentum 6.oldal – 11.oldal

2.feladat a) adatolvasás 12.oldal – 14.oldal

b) adatmódosítás 14.oldal – 23.oldal

c) adatlekérdezés 23.oldal – 28.oldal

d) adatírás 28.oldal – 32.oldal

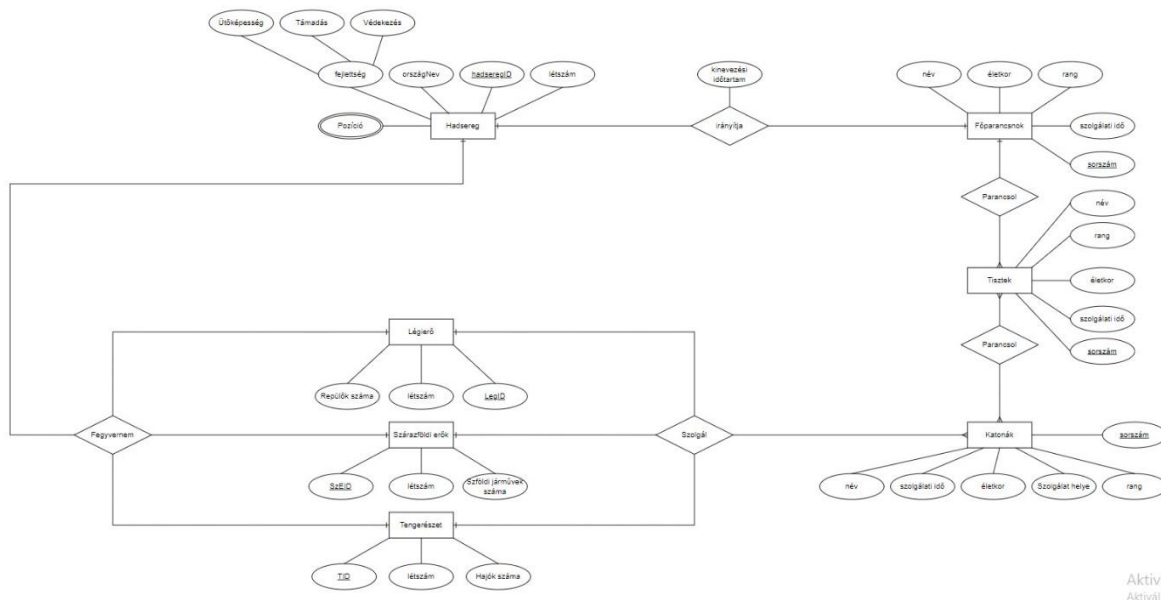
Bevezetés

A féléves beadandóm egy egyszerű hadsereg struktúra bemutatásáról, adatkezeléséről szól.

A már Adatbázisrendszerek I. tantárgy keretein belül elkészített ER modellre alapoztam munkámat. A modell szemlélteti a hadsereget felépítő elemeket, ranglétráit, az egyes rangokhoz tartozó tulajdonságokat, szolgálati helyeket és az ezek között lévő kapcsolatokat. Az XML dokumentum elkészítése előtt, segítségképpen elkészítjük a hozzá tartozó XDM modellt valamint az XML Schema dokumentumot. A létrehozott XML fájl tesztelésére, hasznosságára egy Java projektet hozunk létre, amelyben teszteljük az adatokon végezhető műveleteket. Az adatírás, adatolvasás, adatmódosító és az adatlekérdező programok sikeres lefutásával bizonyosodhatunk meg, hogy XML dokumentumunk felhasználható és sikeresen beépíthető az adott környezetbe.

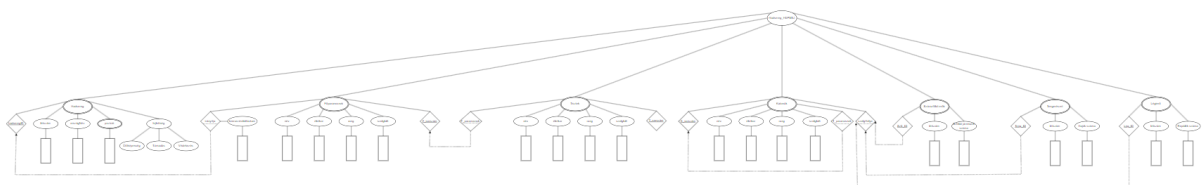
1.feladat: a) Az adatbázis ER modell tervezése

Az itt található képeken található a már korábban elkészített ER modell. A modellen láthatóak a felhasznált egyedek és tulajdonságaik valamint a közöttük lévő kapcsolatok.



b) Adatbázis konvertálása XDM modellre

Itt látható az ER modellről átkonvertált XDM modell. Látható, hogy az ER modellben lévő egyedek egy gyökérellem gyerekei az ábrán. A tulajdonságokon kívül az elsődleges és idegen kulcsok párosítását is szemlélteti az ábra. (PK és FK rombusz alakzatban, szaggatott vonallal összekötve, gyerekelemek kétvonalas ellipszisben, tulajdonságok ellipszisben)



Közelebbi képernyőfelvételek az XDM modellről

c) Az XDM alapján XML dokumentum elkészítése

Az alábbi kódon látható az XML dokumentum elementjei, attribútumai értékekkel feltöltve. A feladatnak megfelelően a többszöri előfordulású elementekből 3 különböző értékekkel rendelkezőt hoztam létre.

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

<Hadsereg_HEPMIU xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
xs:noNamespaceSchemaLocation="XMLSchemaHEPMIU.xsd">

<!--HADSEREG PÉLDÁNYOSÍTÁSA-->

<Hadsereg hadseregID="1">
    <létszám>9000</létszám>
    <országNév>Hungary</országNév>
    <pozíció>Dél-Alföld</pozíció>
    <fejlettség>
```

```
        <ütőképesség>közepes</ütőképesség>
        <támadás>erős</támadás>
        <védekezés>gyenge</védekezés>
    </fejlettség>

</Hadsereg>

<!--FŐPARANCSNOK PÉLDÁNYOSÍTÁSA-->
<Főparancsnok F_sorszám="4545" irányítja="1">
    <név>Nagy József</név>
    <rang>vezérezredes</rang>
    <életkor>56</életkor>
    <szolgIdő>30</szolgIdő>
</Főparancsnok>

<!--TISZTEK PÉLDÁNYOSÍTÁSA-->
<Tisztek T_sorszám="2323" F_parancsnok="4545">
    <név>Király Zoltán</név>
    <rang>százados</rang>
    <életkor>42</életkor>
    <szolgIdő>22</szolgIdő>
</Tisztek>

<Tisztek T_sorszám="2336" F_parancsnok="4545">
    <név>Nagy Zsolt</név>
    <rang>ezredes</rang>
    <életkor>48</életkor>
    <szolgIdő>28</szolgIdő>
</Tisztek>

<Tisztek T_sorszám="9836" F_parancsnok="4545">
    <név>Szabó Gyula</név>
    <rang>őrnagy</rang>
    <életkor>50</életkor>
    <szolgIdő>30</szolgIdő>
</Tisztek>

<!--KATONÁK PÉLDÁNYOSÍTÁSA-->

<Katonák K_sorszám="7874" T_parancsnok="9836" szolgHelye="56">
    <név>Szabó Kristóf</név>
    <rang>tizedes</rang>
    <életkor>20</életkor>
    <szolgIdő>1</szolgIdő>

</Katonák>
```

```

<Katonák K_sorszám="4921" T_parancsnok="9836" szolgHelye="56">
  <név>Tóth Bendegúz</név>
  <rang>hadnagy</rang>
  <életkor>28</életkor>
  <szolgIdő>10</szolgIdő>

</Katonák>

<Katonák K_sorszám="7123" T_parancsnok="9836" szolgHelye="56">
  <név>Tóth Bendegúz</név>
  <rang>tizedes</rang>
  <életkor>23</életkor>
  <szolgIdő>3</szolgIdő>

</Katonák>

<!--FEGYVERNEMEK PÉLDÁNYOSÍTÁSA-->
<Szárzaföldi_erők SzE_ID="56">
  <létszám>5000</létszám>
  <SzföldiJárművekSzám>250</SzföldiJárművekSzám>

</Szárzaföldi_erők>

<Tengerészet Teng_ID="23">
  <létszám>2000</létszám>
  <HajókSzám>5</HajókSzám>

</Tengerészet>

<Légierő Leg_ID="18">
  <létszám>2000</létszám>
  <RepülőKSzám>120</RepülőKSzám>

</Légierő>

</Hadsereg_HEPMIU>

```

d) Az XML dokumentum alapján XML séma elkészítése

Az XML séma dokumentumot a saját típusok meghatározásával kezdtem, majd az elsődleges és idegen kulcsok definiálásával. Utána következik a komplex típusok felsorolása.

```

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

```

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns="http://www.w3.org/2001/XMLSchema">

<!-- SAJÁT TÍPUSOK MEGHATÁROZÁSA-->

<xs:simpleType name="fejlettségTipus">
  <xs:restriction base="xs:string">
    <xs:enumeration value="erős"/>
    <xs:enumeration value="közepes"/>
    <xs:enumeration value="gyenge"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="F_sorszám_típus">
  <xs:restriction base="xs:string">
    <xs:length value="4" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="T_sorszám_típus">
  <xs:restriction base="xs:string">
    <xs:length value="4" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="K_sorszám_típus">
  <xs:restriction base="xs:string">
    <xs:length value="4" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="HadnemID_típus">
  <xs:restriction base="xs:string">
    <xs:length value="3" />
  </xs:restriction>
</xs:simpleType>

<!-- ELSŐDLEGES KULCSOK-->
  <xs:key name="Főparancsnok_kulcsa">
    <xs:selector xpath="Főparancsnok" />
    <xs:field xpath="@F_parancsnok" />
  </xs:key>
```

```
<xs:key name="Hadsereg_kulcsa">
  <xs:selector xpath="Hadsereg" />
  <xs:field xpath="@hadseregID" />
</xs:key>

<xs:key name="Tisztek_kulcsa">
  <xs:selector xpath="Tisztek" />
  <xs:field xpath="@T_sorszám" />
</xs:key>

<xs:key name="Katonák_kulcsa">
  <xs:selector xpath="Katonák" />
  <xs:field xpath="@K_sorszám" />
</xs:key>

<xs:key name="Szárzsföldi_erők_kulcsa">
  <xs:selector xpath="Szárzsföldi_erők" />
  <xs:field xpath="@SzE_ID" />
</xs:key>

  <xs:key name="Légierő_kulcsa">
    <xs:selector xpath="Légierő" />
    <xs:field xpath="@Leg_ID" />
  </xs:key>

  <xs:key name="Tengerészeti_kulcsa">
    <xs:selector xpath="Tengerészeti" />
    <xs:field xpath="@Teng_ID" />
  </xs:key>

<!-- IDEGEN KULCSOK -->
<xs:keyref name="Hadsereg_Főparancsnok_kulcsa" refer="hadseregID">
  <xs:selector xpath="Főparancsnok" />
  <xs:field xpath="@irányítja" />
</xs:keyref>

<xs:keyref name="Főparancsnok_Tisztek_kulcsa" refer="F_sorszám">
  <xs:selector xpath="Tisztek" />
  <xs:field xpath="@parancsol" />
</xs:keyref>

<xs:keyref name="Tisztek_Katonák_kulcsa" refer="T_sorszám">
  <xs:selector xpath="Katonák" />
  <xs:field xpath="@parancsol" />
</xs:keyref>

<xs:keyref name="Ügy_Ítéletet_hoz_kulcs" refer="Ügy_kulcs">
  <xs:selector xpath="Ítéletet_hoz" />
```



```

        <xs:field xpath="@Ügy" />
    </xs:keyref>

    <xs:keyref name="Katonák_Szárazföldi_erők_kulcsa" refer="SzE_ID">
        <xs:selector xpath="Szárazföldi_erők" />
        <xs:field xpath="@szolgál" />
    </xs:keyref>

    <xs:keyref name="Katonák_Tengerészeti_kulcsa" refer="Teng_ID">
        <xs:selector xpath="Tengerészeti" />
        <xs:field xpath="@szolgál" />
    </xs:keyref>

    <xs:keyref name="Katonák_Légierő_kulcsa" refer="Leg_ID">
        <xs:selector xpath="Légierő" />
        <xs:field xpath="@szolgál" />
    </xs:keyref>

<!-- KOMPLEX TÍPUSOK MEGHATÁROZÁSA-->
    <xs:element name="Hadsereg_HEPMIU">
        <xs:complexType>
            <xs:sequence>

<!-- HADSEREG EGYED SÉMÁJA-->
        <xs:element name="Hadsereg">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="létszám" type="xs:int" />
                    <xs:element name="országNév" type="xs:string" />
                    <xs:element name="pozíció" type="xs:string" />
                    <xs:element name="fejlettség">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element name="ütőképesség" type="xs:fejlettségTipus"
/>
                                <xs:element name="támadás" type="xs:fejlettségTipus" />
                                <xs:element name="védekezés" type="xs:fejlettségTipus" />
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                </xs:sequence>
                <xs:attribute name="hadseregID" type="xs:int" use="required" />
            </xs:complexType>
        </xs:element>

<!-- FŐPARANCSNOK EGYED SÉMÁJA-->

```

```

    <xs:element name="Főparancsnok">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="név" type="xs:string" />
          <xs:element name="rang" type="xs:string" />
          <xs:element name="életkor" type="xs:int" />
          <xs:element name="szolgIdő" type="unsignedByte" />
        </xs:sequence>
        <xs:attribute name="F_sorszám" type="F_sorszám_típus"
use="required" />
        <xs:attribute name="irányítja" type="xs:int" use="required" />
      </xs:complexType>
    </xs:element>

<!--TISZTEK EGYED SÉMÁJA-->
    <xs:element maxOccurs="unbounded" name="Tisztek">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="név" type="xs:string" />
          <xs:element name="rang" type="xs:string" />
          <xs:element name="életkor" type="xs:int" />
          <xs:element name="szolgIdő" type="unsignedByte" />
        </xs:sequence>
        <xs:attribute name="T_sorszám" type="T_sorszám_típus"
use="required" />
        <xs:attribute name="F_parancsnok" type="F_sorszám_típus"
use="required" />
      </xs:complexType>
    </xs:element>

<!--KATONÁK EGYED SÉMÁJA-->
    <xs:element maxOccurs="unbounded" name="Katonák">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="név" type="xs:string" />
          <xs:element name="rang" type="xs:string" />
          <xs:element name="életkor" type="xs:int" />
          <xs:element name="szolgIdő" type="unsignedByte" />
        </xs:sequence>
        <xs:attribute name="K_sorszám" type="K_sorszám_típus"
use="required" />
        <xs:attribute name="T_parancsnok" type="T_sorszám_típus"
use="required" />
        <xs:attribute name="szolgHelye" type="HadnemID_típus"
use="required" />

```

```

        </xs:complexType>
    </xs:element>

<!--SZFÖLDI ERŐK EGYED SÉMÁJA-->
    <xs:element name="Szárzsföldi_erők">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="létszám" type="xs:int" />
                <xs:element name="SzföldiJárművekSzáma" type="xs:int" />
            </xs:sequence>
            <xs:attribute name="SzE_ID" type="xs:int" use="required" />
        </xs:complexType>
    </xs:element>

<!--TENGERÉSZET EGYED SÉMÁJA-->
    <xs:element name="Tengerészeti">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="létszám" type="xs:int" />
                <xs:element name="HajókSzáma" type="xs:int" />
            </xs:sequence>
            <xs:attribute name="Teng_ID" type="xs:int" use="required" />
        </xs:complexType>
    </xs:element>

<!--LÉGIERŐ EGYED SÉMÁJA-->
    <xs:element name="Légierő">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="létszám" type="xs:int" />
                <xs:element name="RepülőKszáma" type="xs:int" />
            </xs:sequence>
            <xs:attribute name="Leg_ID" type="xs:int" use="required" />
        </xs:complexType>
    </xs:element>

    </xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

2. feladat: A feladat egy DOM program készítése az XML dokumentum - XMLNeptunkod.xml – adatainak adminisztrálása alapján: (ide kerül a kód - comment együtt)

Project name: DOMParseNeptunkod

Package: hu.domparse.neptunkod

Class names: (DomReadNeptunkod, DomModifyNeptunkod, DomQueryNeptunkod)

2a) adatolvasás

```
package hu.domparse.hepmiu;

import org.w3c.dom.*;

import javax.xml.parsers.*;

import java.io.File;

public class DomReadHepmiu {

    public static void main(String[] args) {

        // TODO Auto-generated method stub

        try {

            File xmlFile = new File("XMLHepmiu.xml");

            DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();

            DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();

            Document doc = dBuilder.parse(xmlFile);

            doc.getDocumentElement().normalize();

            Element gyokerElem = doc.getDocumentElement();

            System.out.println("Gyökér elem: " + gyokerElem.getNodeName());

            kiirTartalom(gyokerElem, "");

        } catch (Exception e) {

            e.printStackTrace();

        }

    }

}
```

```

}

}

private static void kiirTartalom(Node node, String indent) {

    if (node.getNodeType() == Node.ELEMENT_NODE) {

        System.out.println(indent + node.getNodeName());

        if (node.hasAttributes()) {

            NamedNodeMap attrib = node.getAttributes();

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

                Node attribute = attrib.item(i);

                System.out.println(indent + attribute.getNodeName() + " = " +
                    attribute.getNodeValue());

            }

        }

        if (node.hasChildNodes()) {

            NodeList gyerek = node.getChildNodes();

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

                Node child = gyerek.item(i);

                kiirTartalom(child, indent + " ");

            }

        }

    } else if (node.getNodeType() == Node.TEXT_NODE) {

        String datas = node.getNodeValue().trim();

        if (!datas.isEmpty()) {

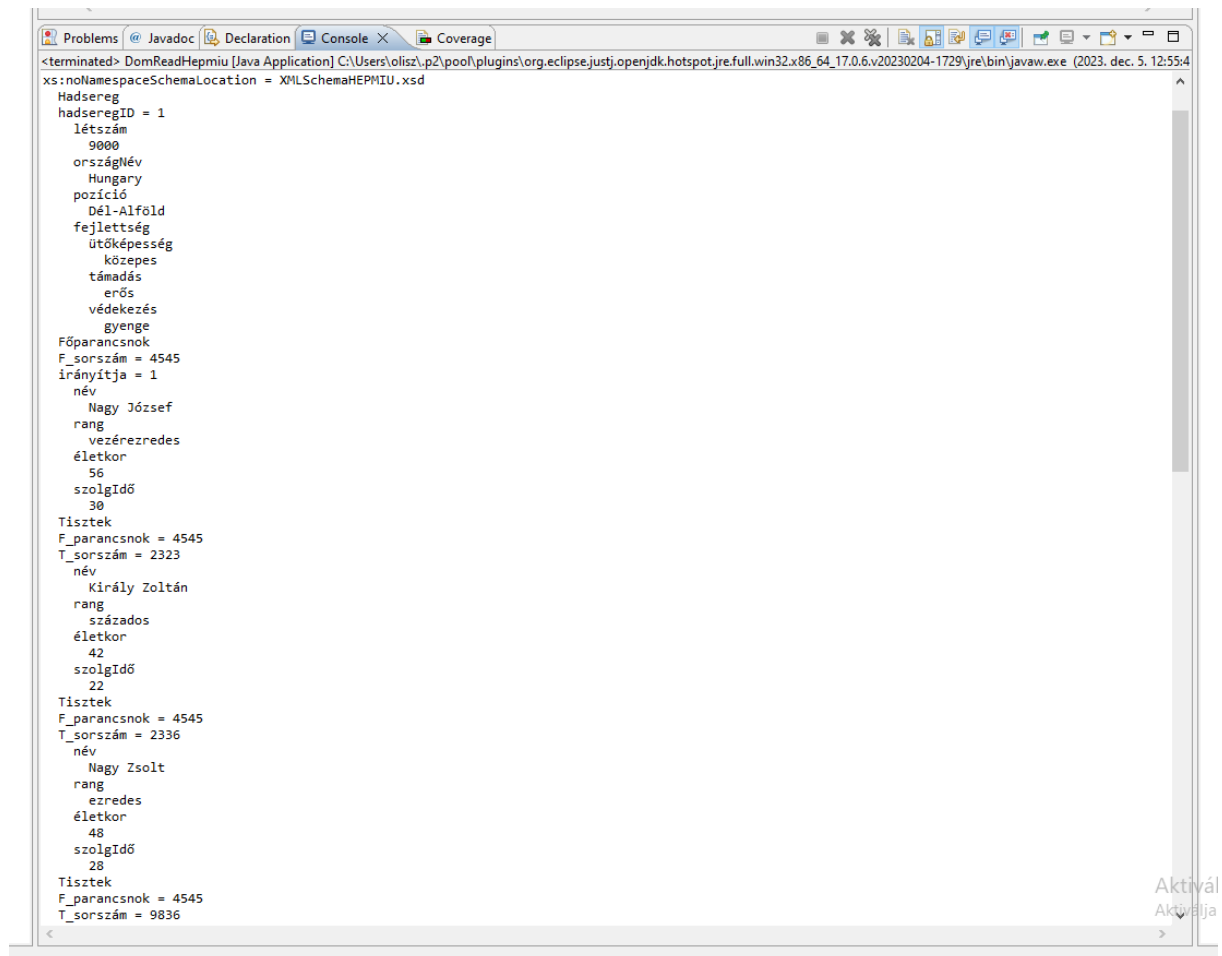
            System.out.println(indent + datas);

        }

    }

}

```



```
<terminated> DomReadHepmiu [Java Application] C:\Users\olisz\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204-1729\jre\bin\javaw.exe (2023. dec. 5. 12:55:4
xs:noNamespaceSchemaLocation = XMLSchemaHEPMIU.xsd
Hadsereg
  hadseregID = 1
  létszám
    9000
  országNév
    Hungary
  pozíció
    Dél-Alföld
  fejlettség
    utöképesség
      közepes
    támadás
      erős
    védekezés
      gyenge
  Főparancsnok
    F_sorszám = 4545
    irányítója = 1
    név
      Nagy József
    rang
      vezérezredes
    életkor
      56
    szolgIdő
      30
  Tisztek
    F_parancsnok = 4545
    T_sorszám = 2323
    név
      Király Zoltán
    rang
      százados
    életkor
      42
    szolgIdő
      22
  Tisztek
    F_parancsnok = 4545
    T_sorszám = 2336
    név
      Nagy Zsolt
    rang
      ezredes
    életkor
      48
    szolgIdő
      28
  Tisztek
    F_parancsnok = 4545
    T_sorszám = 9836
```

Aktivál
Aktiválj

2b) adاتمódosítás

```
package hu.domparse.hepmiu;

import org.w3c.dom.*;

import javax.xml.parsers.*;

import javax.xml.transform.*;

import javax.xml.transform.dom.DOMSource;

import javax.xml.transform.stream.StreamResult;

import java.io.File;

import java.util.Scanner;

public class DomModifyHepmiu {

    public static void main(String[] args) {

        try {

            File xmlFile = new File("XMLHepmiu.xml");
```

```

DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();

DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();

Document doc = dBuilder.parse(xmlFile);

doc.getDocumentElement().normalize();

Scanner sc = new Scanner(System.in);

System.out.println("Adja meg a módosítani kívánt element nevét!");

String elementName = sc.nextLine();

System.out.println("Adja meg a módosítani kívánt element ID-ját!");

String elementID = sc.nextLine();

System.out.println("Adja meg a módosítani kívánt element attribútumát vagy
gyerekének nevét!");

String attributeNameOrChildName = sc.nextLine();

System.out.println("Adja meg az új értékét");

String newValue = sc.nextLine();

modifyElementByID(doc, elementName, elementID, attributeNameOrChildName,
newValue);

sc.close();

writeToFile(doc, "XMLHepmiu1.xml");

System.out.println("Adat sikeresen módosítva!");

} catch (Exception e) {

e.printStackTrace();

}

}

public static void modifyElementByID(Document doc, String elementName, String
elementID,

String attributeNameOrChildName, String newValue) {

NodeList nodeList = doc.getElementsByTagName(elementName);

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

Node node = nodeList.item(i);

```

```

if (node.getNodeType() == Node.ELEMENT_NODE) {

    Element element = (Element) node;

    if (elementName.equalsIgnoreCase("Katonák") ||
        elementName.equalsIgnoreCase("Tisztek")

        || elementName.equalsIgnoreCase("Főparancsnok")) {

        if (element.getAttribute(elementName.charAt(0) + "_sorszám").equals(elementID)) {

            if (element.hasAttribute(attributeNameOrChildName)) {

                element.setAttribute(attributeNameOrChildName, newValue);

            } else {

                NodeList childNodes = element.getElementsByTagName(attributeNameOrChildName);

                if (childNodes.getLength() > 0) {

                    Node childNode = childNodes.item(0);

                    childNode.setTextContent(newValue);

                } else {

                    System.out.println("Adat típus nem található: " + attributeNameOrChildName);

                }

            }

        }

        else if (elementName.equalsIgnoreCase("Szárzsföldi_erők") ||
            elementName.equalsIgnoreCase("Tengerészeti")

            || elementName.equalsIgnoreCase("Légierő")) {

            if (element.getAttribute("SzE_ID").equals(elementID)

                || element.getAttribute("Teng_ID").equals(elementID)

                || element.getAttribute("Leg_ID").equals(elementID)) {

                if (element.hasAttribute(attributeNameOrChildName)) {

                    element.setAttribute(attributeNameOrChildName, newValue);

                } else {

```



```

NodeList childNodes = element.getElementsByTagName(attributeNameOrChildName);

if (childNodes.getLength() > 0) {
    Node childNode = childNodes.item(0);
    childNode.setTextContent(newValue);
} else {
    System.out.println("Adat típus nem található: " + attributeNameOrChildName);
}

}

}

}

else if (elementName.equalsIgnoreCase("Hadsereg")) {
    if (element.getAttribute("hadseregID").equals(elementID)) {
        if (element.hasAttribute(attributeNameOrChildName)) {
            element.setAttribute(attributeNameOrChildName, newValue);
        } else {
            NodeList childNodes = element.getElementsByTagName(attributeNameOrChildName);

            if (childNodes.getLength() > 0) {
                Node childNode = childNodes.item(0);
                childNode.setTextContent(newValue);
            } else {
                System.out.println("Adat típus nem található: " + attributeNameOrChildName);
            }
        }
    }
}
}
}
}
}
}
}
}

```

```

}

public static void writeToFile(Document doc, String filename) {

try {

TransformerFactory transformerFactory = TransformerFactory.newInstance();

Transformer transformer = transformerFactory.newTransformer();

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

DOMSource source = new DOMSource(doc);

StreamResult result = new StreamResult(new File(filename));

transformer.transform(source, result);

} catch (Exception e) {

e.printStackTrace();

}

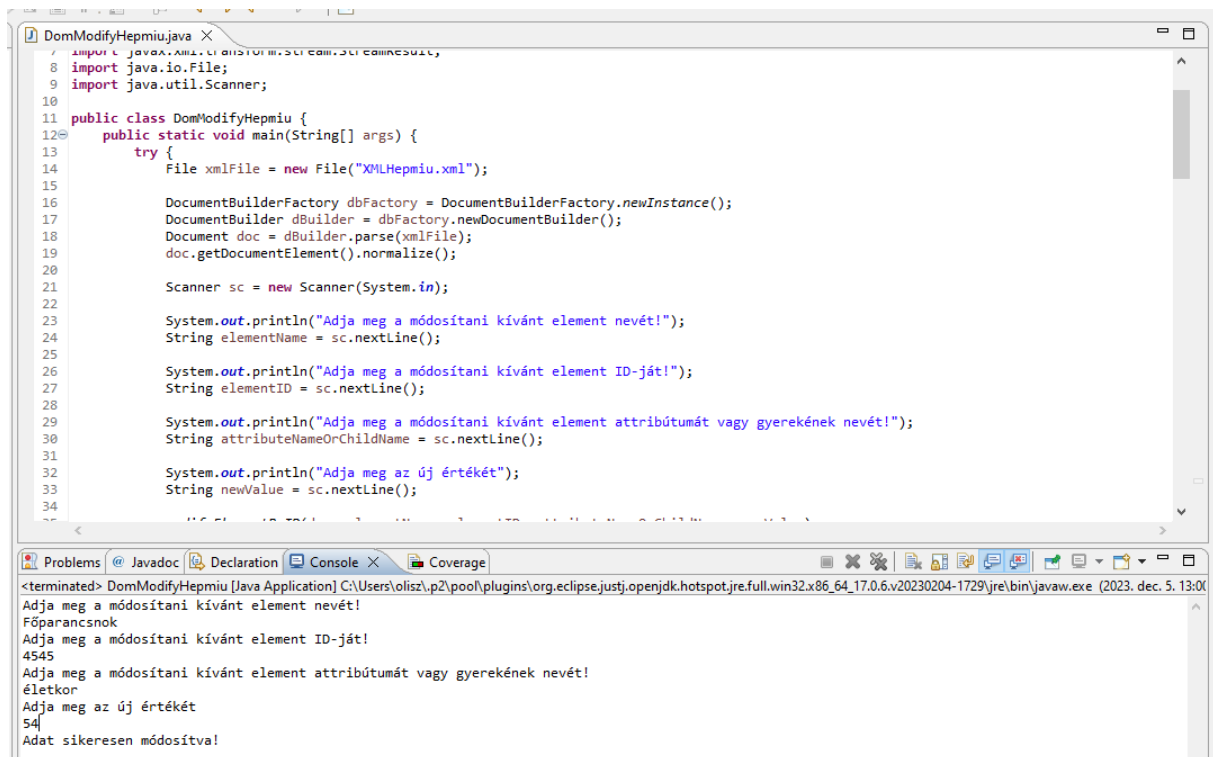
}

}

}

```

A főparancsnok életkorának módosítása:



The screenshot shows the Eclipse IDE with the file `DomModifyHepmiu.java` open. The code in the editor is as follows:

```

1 import java.xml.transform.stream.StreamResult;
2 import java.io.File;
3 import java.util.Scanner;
4
5 public class DomModifyHepmiu {
6     public static void main(String[] args) {
7         try {
8             File xmlFile = new File("XMLHepmiu.xml");
9
10            DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
11            DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
12            Document doc = dBuilder.parse(xmlFile);
13            doc.getDocumentElement().normalize();
14
15            Scanner sc = new Scanner(System.in);
16
17            System.out.println("Adja meg a módosítani kívánt element nevét!");
18            String elementName = sc.nextLine();
19
20            System.out.println("Adja meg a módosítani kívánt element ID-ját!");
21            String elementID = sc.nextLine();
22
23            System.out.println("Adja meg a módosítani kívánt element attribútumát vagy gyerekének nevét!");
24            String attributeNameOrChildName = sc.nextLine();
25
26            System.out.println("Adja meg az új értéket");
27            String newValue = sc.nextLine();
28
29            // ... (the rest of the code is obscured by a watermark)
30        }
31    }
32 }

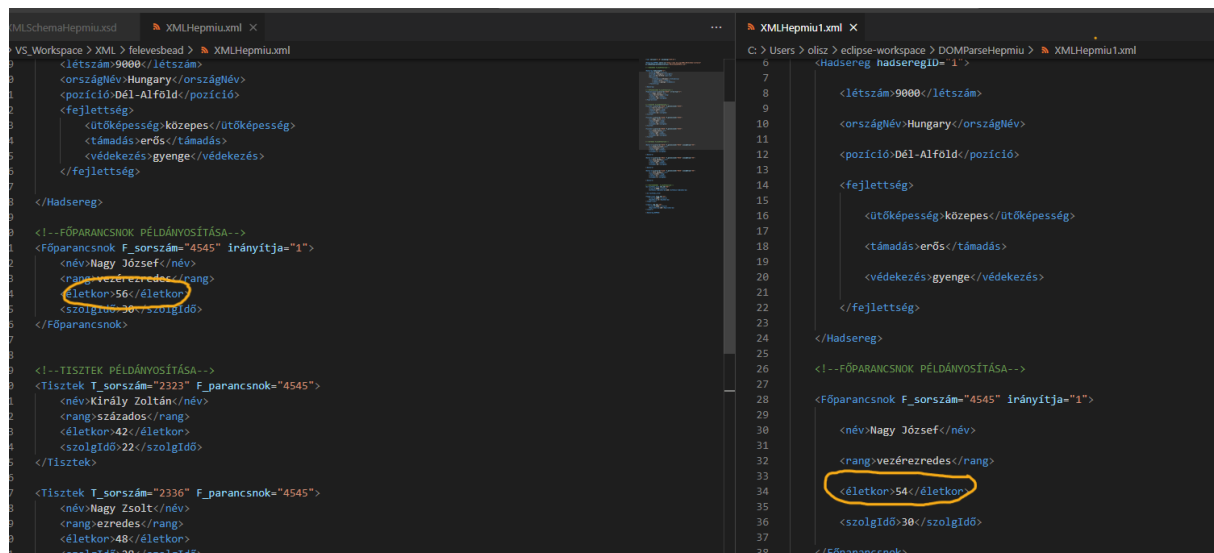
```

The console output at the bottom shows the execution of the program:

```

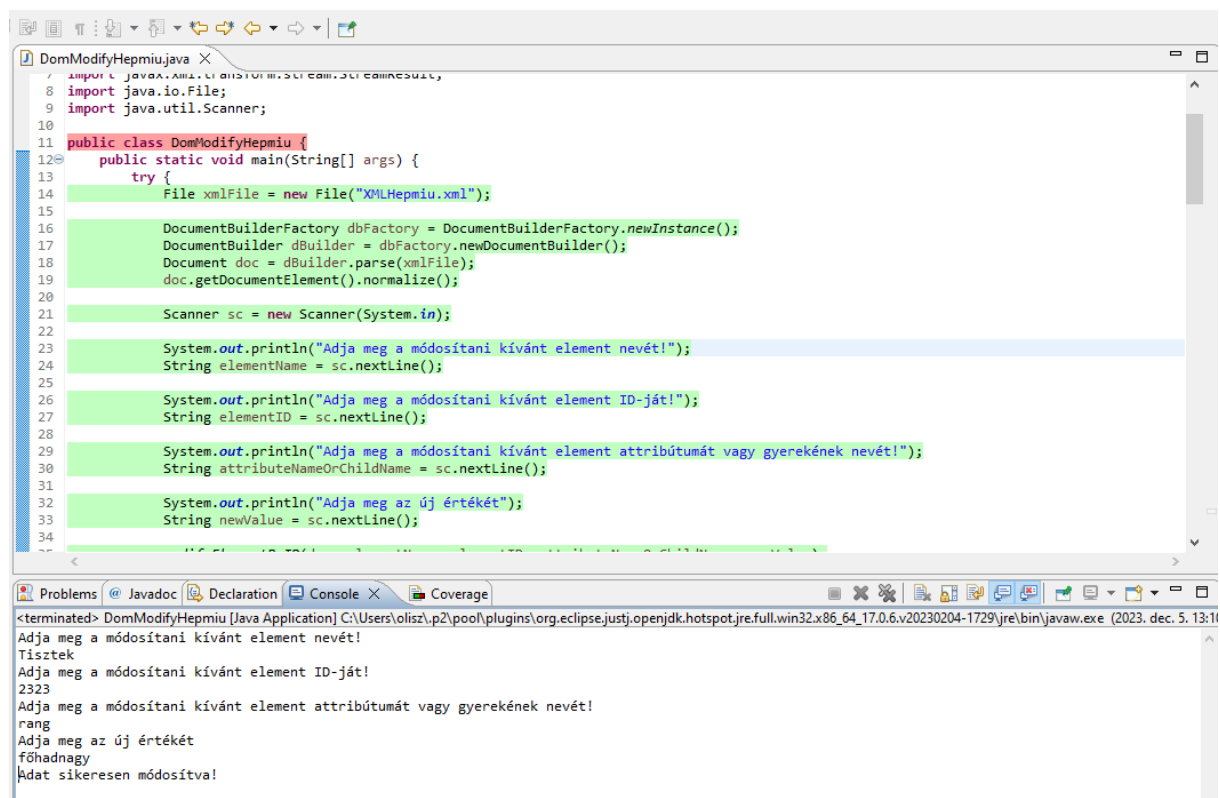
<terminated> DomModifyHepmiu [Java Application] C:\Users\olisz\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204-1729\jre\bin\javaw.exe (2023. dec. 5. 13:00)
Adja meg a módosítani kívánt element nevét!
Főparancsnok
Adja meg a módosítani kívánt element ID-ját!
4545
Adja meg a módosítani kívánt element attribútumát vagy gyerekének nevét!
életkor
Adja meg az új értéket
54
Adat sikeresen módosítva!

```



„2323” -as sorszámú Tiszt rangjának módosítása századosról főhadnagyra.

Eclipse IDE



```
<!--TISZTEK PÉLDÁNYOSÍTÁSA-->
<Tisztek T_sorszám="2323" F_parancsnok="4545">
  <név>Király Zoltán</név>
  <rang>százados</rang>
  <életkor>42</életkor>
  <szolgIdő>22</szolgIdő>
</Tisztek>

<Tisztek T_sorszám="2336" F_parancsnok="4545">
  <név>Nagy Zsolt</név>
  <rang>ezredes</rang>
  <életkor>48</életkor>
  <szolgIdő>28</szolgIdő>
</Tisztek>

<Tisztek T_sorszám="9836" F_parancsnok="4545">
  <név>Szabó Gyula</név>
  <rang>őrnagy</rang>
  <életkor>50</életkor>
  <szolgIdő>30</szolgIdő>
</Tisztek>

29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
<név>Nagy József</név>
<rang>vezérezredes</rang>
<életkor>56</életkor>
<szolgIdő>30</szolgIdő>
</Főparancsnok>

<!--TISZTEK PÉLDÁNYOSÍTÁSA-->
<Tisztek F_parancsnok="4545" T_sorszám="2323">
  <név>Király Zoltán</név>
  <rang>főhadnagy</rang>
  <életkor>42</életkor>
  <szolgIdő>22</szolgIdő>
</Tisztek>
```

„9836” sorszámú Tiszt nevének módosítása Szabó Gyuláról Szabó Györgyre.

- Eclipse IDE

The screenshot shows the Eclipse IDE with the file `DomModifyHepmiu.java` open. The code is a Java application that reads XML data, prompts the user for modifications, and updates the XML. The console output shows the program's execution, including prompts for element name, ID, attribute name, and new value, followed by the successful update of the XML data.

```
DomModifyHepmiu.java
1 import java.io.*;
2 import java.io.File;
3 import java.util.Scanner;
4
5 public class DomModifyHepmiu {
6     public static void main(String[] args) {
7         try {
8             File xmlFile = new File("XMLHepmiu.xml");
9
10            DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
11            DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
12            Document doc = dBuilder.parse(xmlFile);
13            doc.getDocumentElement().normalize();
14
15            Scanner sc = new Scanner(System.in);
16
17            System.out.println("Adja meg a módosítani kívánt element nevét!");
18            String elementName = sc.nextLine();
19
20            System.out.println("Adja meg a módosítani kívánt element ID-ját!");
21            String elementID = sc.nextLine();
22
23            System.out.println("Adja meg a módosítani kívánt element attribútumát vagy gyerekének nevét!");
24            String attributeNameOrChildName = sc.nextLine();
25
26            System.out.println("Adja meg az új értékét");
27            String newValue = sc.nextLine();
28
29            // ... (code for updating the XML) ...
30
31        } catch (Exception e) {
32            e.printStackTrace();
33        }
34    }
35 }
```

Problems | Javadoc | Declaration | Console | Coverage

```
<terminated> DomModifyHepmiu [Java Application] C:\Users\olisz\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204-1729\jre\bin\javaw.exe (2023. dec. 5. 13:16)
Adja meg a módosítani kívánt element nevét!
Tisztek
Adja meg a módosítani kívánt element ID-ját!
9836
Adja meg a módosítani kívánt element attribútumát vagy gyerekének nevét!
név
Adja meg az új értékét
Szabó György
Adat sikeresen módosítva!
```

```
<szolgIdő>22</szolgIdő>
</Tisztek>

<Tisztek T_sorszám="2336" F_parancsnok="4545">
  <név>Nagy Zsolt</név>
  <rang>ezredes</rang>
  <életkor>48</életkor>
  <szolgIdő>28</szolgIdő>
</Tisztek>

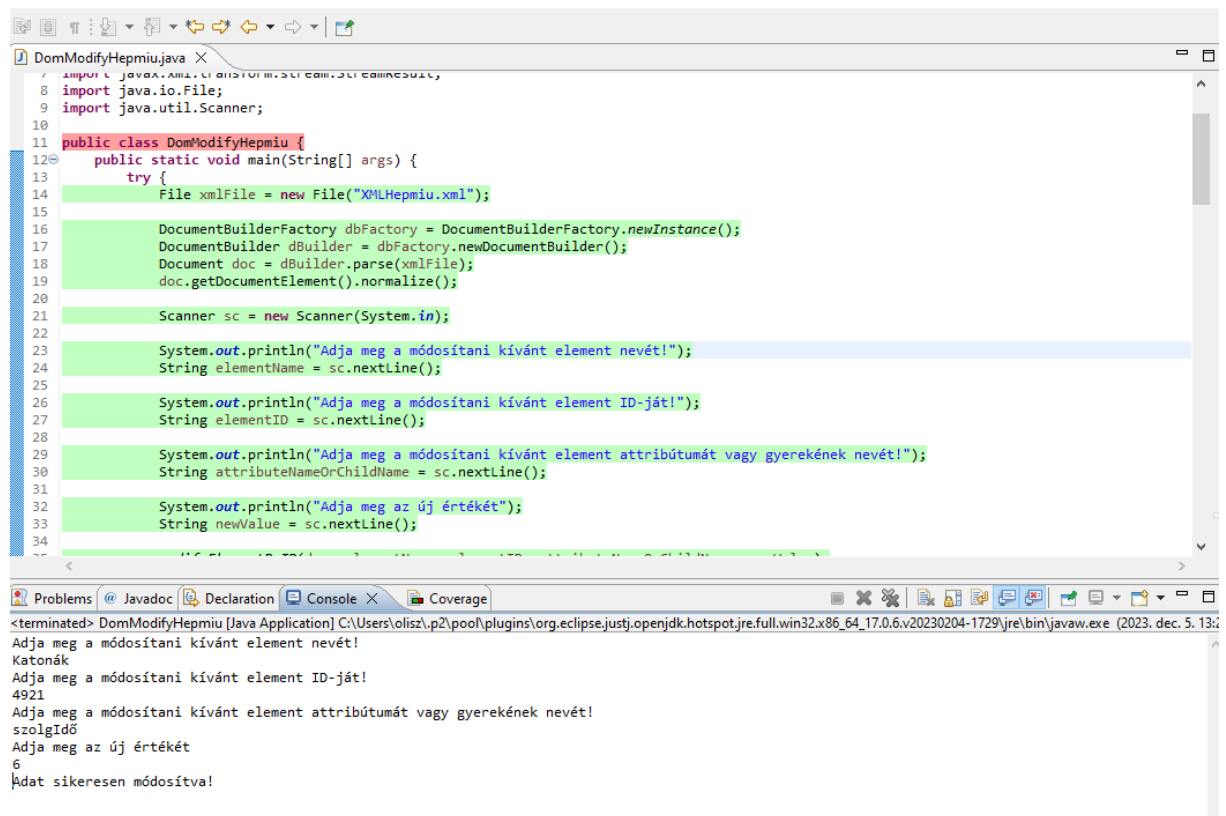
<Tisztek T_sorszám="9836" F_parancsnok="4545">
  <név>Szabó Gyula</név>
  <rang>őrnagy</rang>
  <életkor>50</életkor>
  <szolgIdő>30</szolgIdő>
</Tisztek>

60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
<életkor>48</életkor>
<szolgIdő>28</szolgIdő>
</Tisztek>

<Tisztek F_parancsnok="4545" T_sorszám="9836">
  <név>Szabó György</név>
  <rang>őrnagy</rang>
  <életkor>50</életkor>
  <szolgIdő>30</szolgIdő>
</Tisztek>
```

„4921” sorszámú Katona Szolgálati idejének megváltoztatása 10-ről 6-ra.

Eclipse IDE

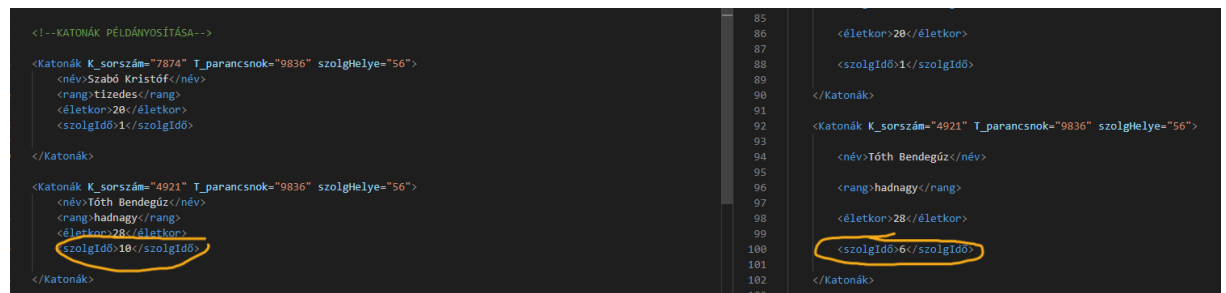


```
1 import java.io.IOException;
2 import java.io.File;
3 import java.util.Scanner;
4
5 public class DomModifyHepmiu {
6     public static void main(String[] args) {
7         try {
8             File xmlFile = new File("XMLHepmiu.xml");
9
10            DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
11            DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
12            Document doc = dBuilder.parse(xmlFile);
13            doc.getDocumentElement().normalize();
14
15            Scanner sc = new Scanner(System.in);
16
17            System.out.println("Adja meg a módosítani kívánt element nevét!");
18            String elementName = sc.nextLine();
19
20            System.out.println("Adja meg a módosítani kívánt element ID-ját!");
21            String elementID = sc.nextLine();
22
23            System.out.println("Adja meg a módosítani kívánt element attribútumát vagy gyerekének nevét!");
24            String attributeNameOrChildName = sc.nextLine();
25
26            System.out.println("Adja meg az új értékét");
27            String newValue = sc.nextLine();
28
29            // ... (rest of the code) ...
30        }
31    }
32}
```

Problems | Javadoc | Declaration | Console | Coverage

<terminated> DomModifyHepmiu [Java Application] C:\Users\olisz\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204-1729\jre\bin\javaw.exe (2023. dec. 5. 13:00)

Adja meg a módosítani kívánt element nevét!
Katonák
Adja meg a módosítani kívánt element ID-ját!
4921
Adja meg a módosítani kívánt element attribútumát vagy gyerekének nevét!
szolgIdő
Adja meg az új értékét
6
Adat sikeresen módosítva!



```
<!--KATONÁK PÉLDÁNYOSÍTÁSA-->
<Katonák K_sorszám="7874" T_parancsnok="9836" szolgHelye="56">
  <név>Szabó Kristóf</név>
  <rang>tizedes</rang>
  <életkor>20</életkor>
  <szolgIdő>1</szolgIdő>
</Katonák>

<Katonák K_sorszám="4921" T_parancsnok="9836" szolgHelye="56">
  <név>Tóth Bendegúz</név>
  <rang>hadnagy</rang>
  <életkor>28</életkor>
  <szolgIdő>10</szolgIdő>
</Katonák>
```

```
85
86 <életkor>20</életkor>
87
88 <szolgIdő>1</szolgIdő>
89
90 </Katonák>
91
92 <Katonák K_sorszám="4921" T_parancsnok="9836" szolgHelye="56">
93
94 <név>Tóth Bendegúz</név>
95
96 <rang>hadnagy</rang>
97
98 <életkor>28</életkor>
99 <szolgIdő>6</szolgIdő>
100
101 </Katonák>
102
103
```

A Szárazföldi erők element létszámának növelése 5000-ről 6500-ra.

```

1  import javax.xml.parsers.*;
2  import java.io.File;
3  import java.util.Scanner;
4
5  public class DomModifyHepmiu {
6      public static void main(String[] args) {
7          try {
8              File xmlFile = new File("XMLHepmiu.xml");
9
10             DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
11             DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
12             Document doc = dBuilder.parse(xmlFile);
13             doc.getDocumentElement().normalize();
14
15             Scanner sc = new Scanner(System.in);
16
17             System.out.println("Adja meg a módosítani kívánt element nevét!");
18             String elementName = sc.nextLine();
19
20             System.out.println("Adja meg a módosítani kívánt element ID-ját!");
21             String elementID = sc.nextLine();
22
23             System.out.println("Adja meg a módosítani kívánt element attribútumát vagy gyerekének nevét!");
24             String attributeNameOrChildName = sc.nextLine();
25
26             System.out.println("Adja meg az új értékét");
27             String newValue = sc.nextLine();
28
29             // ... (code for modifying the XML) ...
30
31         } catch (Exception e) {
32             e.printStackTrace();
33         }
34     }
35 }

```

Problems | Javadoc | Declaration | Console | Coverage

```

<terminated> DomModifyHepmiu [Java Application] C:\Users\olisz\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204-1729\jre\bin\javaw.exe (2023. dec. 5. 13:24)
Adja meg a módosítani kívánt element nevét!
Százazföldi_erők
Adja meg a módosítani kívánt element ID-ját!
56
Adja meg a módosítani kívánt element attribútumát vagy gyerekének nevét!
létszám
Adja meg az új értékét
6500
Adat sikeresen módosítva!

```

```

71  <név>Tóth Bendegúz</név>
72  <rang>tizedes</rang>
73  <életkor>23</életkor>
74  <szolgIdő>3</szolgIdő>
75
76  </Katonák>
77
78
79  <!--FEGVERNEK PÉLDÁNYOSÍTÁSA-->
80  <Százazföldi_erők SzE_ID="56">
81      <létszám>5000</létszám>
82      <SzföldiJárművekSzám>250</SzföldiJárművekSzám>
83  </Százazföldi_erők>
84
85  <Tengerészeti_Teng_ID="23">
86      <létszám>2000</létszám>
87      <HajókSzám>5</HajókSzám>
88  </Tengerészeti_Teng_ID="23">
89
90  <Légi_erők Leg_ID="18">
91      <létszám>2000</létszám>
92      <RepülőKészletSzám>120</RepülőKészletSzám>
93
115
116  <!--FEGVERNEK PÉLDÁNYOSÍTÁSA-->
117
118  <Százazföldi_erők SzE_ID="56">
119      <létszám>6500</létszám>
120      <SzföldiJárművekSzám>250</SzföldiJárművekSzám>
121  </Százazföldi_erők>
122
123  <Tengerészeti_Teng_ID="23">
124      <létszám>2000</létszám>
125      <HajókSzám>5</HajókSzám>
126  </Tengerészeti_Teng_ID="23">
127
128  <Légi_erők Leg_ID="18">
129      <létszám>2000</létszám>
130      <RepülőKészletSzám>120</RepülőKészletSzám>
131
132  </Légi_erők Leg_ID="18">
133
134  </FEGVERNEK PÉLDÁNYOSÍTÁSA-->
135
136  </FEGVERNEK PÉLDÁNYOSÍTÁSA-->
137
138  </FEGVERNEK PÉLDÁNYOSÍTÁSA-->

```

2c) adat lekérdezés

package hu.domparse.hepmiu;

import org.w3c.dom.*;

import javax.xml.parsers.*;

import java.io.File;

import java.util.Scanner;

public class DomQueryHepmiu {

public static void main(String[] args) {

try {

```

File xmlFile = new File("XMLHepmiu.xml");

DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();

DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();

Document doc = dBuilder.parse(xmlFile);

doc.getDocumentElement().normalize();

//LEKÉRDEZNI KÍVÁNT ELEMENT LEKÉRDEZÉSE

Scanner sc = new Scanner(System.in);

System.out.println("Adja meg az element nevet!");

String elementName = sc.nextLine();

System.out.println("Adja meg az element ID-janak ertekeit!");

String elementID = sc.nextLine();

sc.close();

selectElementByID(doc, elementName, elementID);

} catch (Exception e) {

e.printStackTrace();

}

}

public static void selectElementByID(Document doc, String elementName, String
elementID) {

NodeList nodeList = doc.getElementsByTagName(elementName);

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

Node node = nodeList.item(i);

if (node.getNodeType() == Node.ELEMENT_NODE) {

Element element = (Element) node;

if (elementName.equalsIgnoreCase("Katonák") ||
elementName.equalsIgnoreCase("Tisztek")

|| elementName.equalsIgnoreCase("Főparancsnok")) {

if (element.getAttribute(elementName.charAt(0) + "_sorszám").equals(elementID)) {

```

```
// ELEMENTEK KIÍRÁSA ÉRTÉKEIKKEL EGYÜTT
```

```
System.out.println("Element Neve: " + element.getNodeName());

NamedNodeMap attributes = element.getAttributes();

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

    Node attribute = attributes.item(j);

    System.out.println(attribute.getNodeName() + ": " + attribute.getNodeValue());

}

NodeList children = element.getChildNodes();

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

    Node child = children.item(k);

    if (child.getNodeType() == Node.ELEMENT_NODE) {

        System.out.println(child.getNodeName() + ": " + child.getTextContent());

    }

}

}

}

else if (elementName.equalsIgnoreCase("Szárazföldi_erők") ||
elementName.equalsIgnoreCase("Tengerészeti")) {

    || elementName.equalsIgnoreCase("Légierő")) {

        if (element.getAttribute("SzE_ID").equals(elementID)

        || element.getAttribute("Teng_ID").equals(elementID)

        || element.getAttribute("Leg_ID").equals(elementID)) {

// ELEMENTEK KIÍRÁSA ÉRTÉKEIKKEL EGYÜTT

System.out.println("Element Neve: " + element.getNodeName());

NamedNodeMap attributes = element.getAttributes();

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

    Node attribute = attributes.item(j);

    System.out.println(attribute.getNodeName() + ": " + attribute.getNodeValue());
```



```

}

NodeList children = element.getChildNodes();

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

Node child = children.item(k);

if (child.getNodeType() == Node.ELEMENT_NODE) {

System.out.println(child.getNodeName() + ": " + child.getTextContent());

}

}

}

}

else if (elementName.equalsIgnoreCase("Hadsereg")) {

if (element.getAttribute("hadseregID").equals(elementID)) {

// ELEMENTEK KIÍRÁSA ÉRTÉKEIKKEL EGYÜTT

System.out.println("Element Neve: " + element.getNodeName());

NamedNodeMap attributes = element.getAttributes();

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

Node attribute = attributes.item(j);

System.out.println(attribute.getNodeName() + ": " + attribute.getNodeValue());

}

NodeList children = element.getChildNodes();

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

Node child = children.item(k);

if (child.getNodeType() == Node.ELEMENT_NODE) {

System.out.println(child.getNodeName() + ": " + child.getTextContent());

}

}

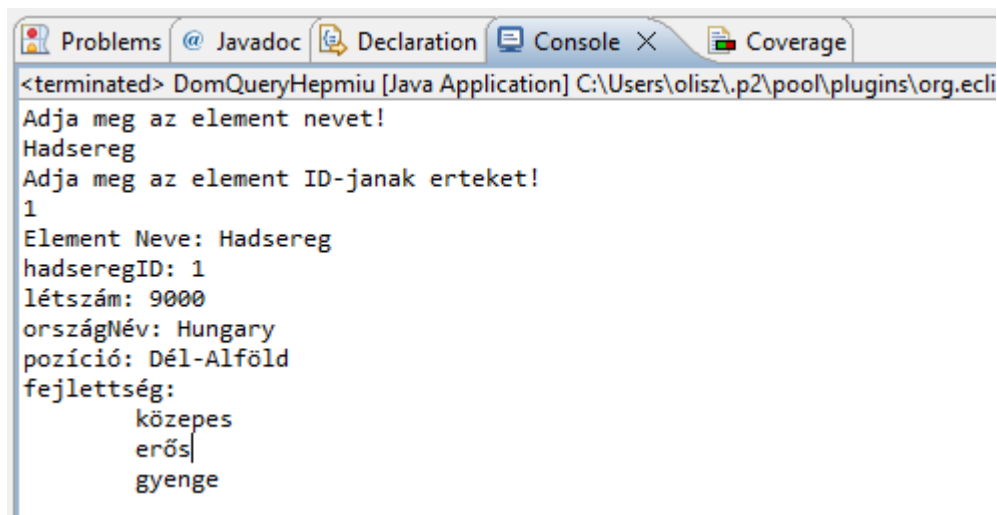
}

}

```

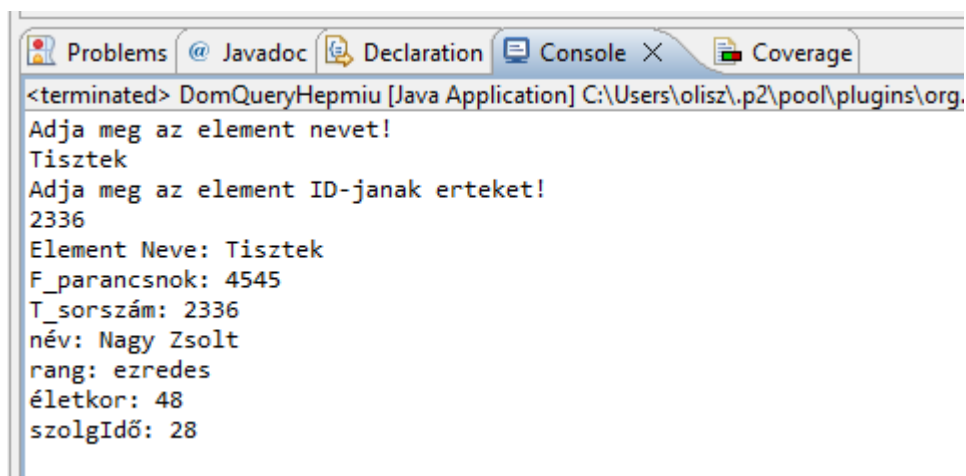
```
}  
}  
}  
}  
}  
}  
}  
}  
}  
}
```

Hadsereg element lekérdezése



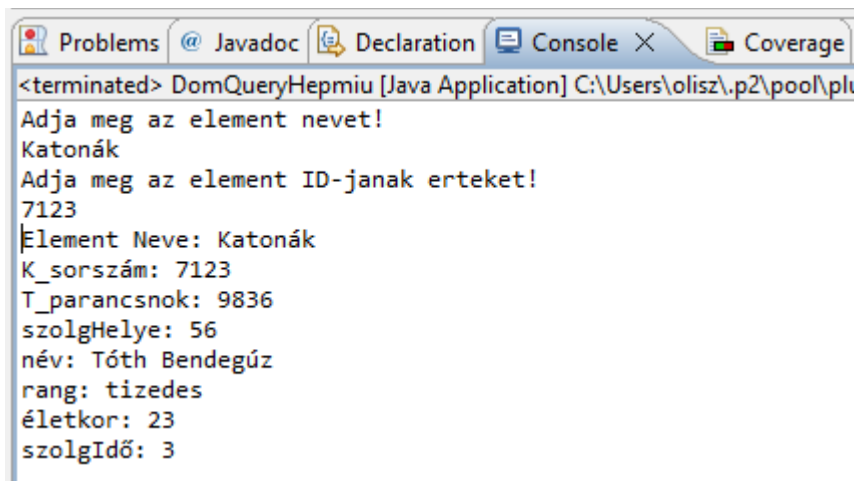
```
<terminated> DomQueryHepmiu [Java Application] C:\Users\olisz\.p2\pool\plugins\org.ecl  
Adja meg az element nevet!  
Hadsereg  
Adja meg az element ID-janak erteket!  
1  
Element Neve: Hadsereg  
hadseregID: 1  
létszám: 9000  
országNév: Hungary  
pozíció: Dél-Alföld  
fejlettség:  
    közepes  
    erős  
    gyenge
```

„2336” sorszámú Tiszt adatainak lekérdezése



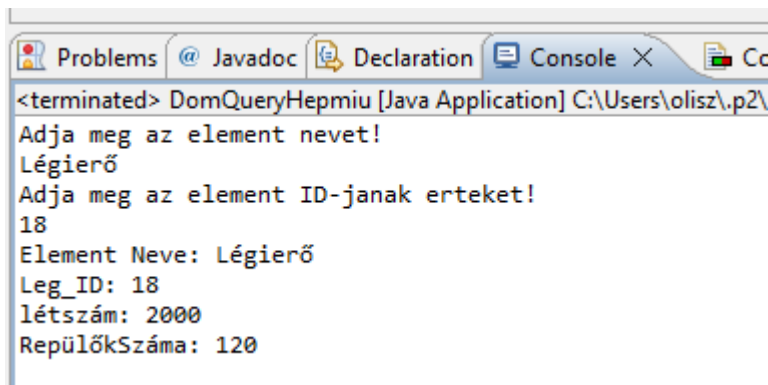
```
<terminated> DomQueryHepmiu [Java Application] C:\Users\olisz\.p2\pool\plugins\org.  
Adja meg az element nevet!  
Tisztek  
Adja meg az element ID-janak erteket!  
2336  
Element Neve: Tisztek  
F_parancsnok: 4545  
T_sorszám: 2336  
név: Nagy Zsolt  
rang: ezredes  
életkor: 48  
szolgIdő: 28
```

„7123” sorszámú Katona lekérdezése



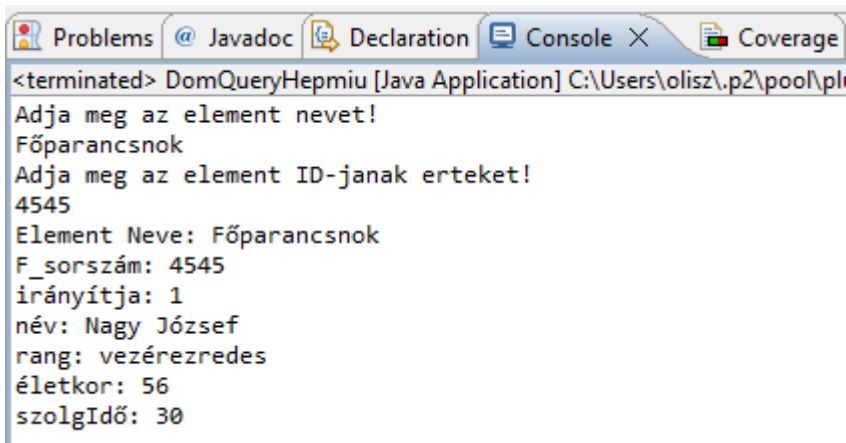
```
Problems  Javadoc  Declaration  Console  Coverage
<terminated> DomQueryHepmiu [Java Application] C:\Users\olisz\.p2\pool\pl
Adja meg az element nevet!
Katonák
Adja meg az element ID-janak erteket!
7123
Element Neve: Katonák
K_sorszám: 7123
T_parancsnok: 9836
szolgHelye: 56
név: Tóth Bendegúz
rang: tizedes
életkor: 23
szolgIdő: 3
```

Légierő element adatainak lekérdezése



```
Problems  Javadoc  Declaration  Console  Coverage
<terminated> DomQueryHepmiu [Java Application] C:\Users\olisz\.p2\
Adja meg az element nevet!
Légierő
Adja meg az element ID-janak erteket!
18
Element Neve: Légierő
Leg_ID: 18
létszám: 2000
RepülőKszáma: 120
```

Főparancsnok adatainak lekérdezése



```
Problems  Javadoc  Declaration  Console  Coverage
<terminated> DomQueryHepmiu [Java Application] C:\Users\olisz\.p2\pool\pl
Adja meg az element nevet!
Főparancsnok
Adja meg az element ID-janak erteket!
4545
Element Neve: Főparancsnok
F_sorszám: 4545
irányítja: 1
név: Nagy József
rang: vezérezredes
életkor: 56
szolgIdő: 30
```

2d) adatírás

```
package hu.domparse.hepmiu;
```

```
import java.io.File;
```

```

import java.io.IOException;

import javax.xml.parsers.ParserConfigurationException;

import javax.xml.parsers.SAXParser;

import javax.xml.parsers.SAXParserFactory;

import org.xml.sax.Attributes;

import org.xml.sax.SAXException;

import org.xml.sax.helpers.DefaultHandler;

import org.w3c.dom.*;

import javax.xml.parsers.*;

import javax.xml.transform.*;

import javax.xml.transform.dom.DOMSource;

import javax.xml.transform.stream.StreamResult;

import java.io.File;

public class DOMWriteHepmiu {

    public static void main(String[] args) {

        try {

            File xmlFile = new File("XMLHepmiu.xml");

            DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();

            DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();

            Document doc = dBuilder.parse(xmlFile);

            doc.getDocumentElement().normalize();

            Element gyokerElem = doc.getDocumentElement();

            System.out.println("Gyökér elem: " + gyokerElem.getNodeName());

            kiirTartalom(gyokerElem, "");

            writeToFile(doc, "XMLHepmiu1.xml"); // UJ XML FAJLT LETREHOZZA

            System.out.println("XML updatetelt változata mentve az XMLHepmiu1.xml fájlba");

        } catch (Exception e) {

```

```

e.printStackTrace();

}

}

private static void kiirTartalom(Node node, String indent) {

    if (node.getNodeType() == Node.ELEMENT_NODE) {

        System.out.println(indent + node.getNodeName());

        if (node.hasAttributes()) {

            NamedNodeMap attrib = node.getAttributes();

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

                Node attribute = attrib.item(i);

                System.out.println(indent + attribute.getNodeName() + " = " +
                    attribute.getNodeValue());

            }

        }

        if (node.hasChildNodes()) {

            NodeList gyerek = node.getChildNodes();

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

                Node child = gyerek.item(i);

                kiirTartalom(child, indent + " ");

            }

        }

    } else if (node.getNodeType() == Node.TEXT_NODE) {

        String datas = node.getNodeValue().trim();

        if (!datas.isEmpty()) {

            System.out.println(indent + datas);

        }

    }

}

```

```

public static void writeToFile(Document doc, String filename) { // A MODOSÍTOTT
FAJLT LETREHOZO METODUS

try {

TransformerFactory transformerFactory = TransformerFactory.newInstance();

Transformer transformer = transformerFactory.newTransformer();

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

DOMSource source = new DOMSource(doc);

StreamResult result = new StreamResult(new File(filename));

transformer.transform(source, result);

} catch (Exception e) {

e.printStackTrace();

}

}









}

```

```

<terminated> DOMWriteHepmiu [Java Application] C:\Users\olisz\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204-1729\jre\bin\javaw.exe (2023. dec. 5. 13:40:
K_sorszám = 4921
T_parancsnok = 9836
szolgHelye = 56
név
Tóth Bendegúz
rang
hadnagy
életkor
28
szolgIdő
10
Katonák
K_sorszám = 7123
T_parancsnok = 9836
szolgHelye = 56
név
Tóth Bendegúz
rang
tizedes
életkor
23
szolgIdő
3
Százfőldi_erők
SzE_ID = 56
létszám
6500
SzföldiJárművekSzám
250
Tengerész
Teng_ID = 23
létszám
2000
HajókSzám
5
Légierő
Leg_ID = 18
létszám
2000
RepülőSzám
120
XML adatetelt változata mentve az XMLHepmiu2.xml fájlba

```

 .settings	2023. 11. 04. 13:48	Fájlmappa	
 bin	2023. 11. 29. 11:26	Fájlmappa	
 src	2023. 11. 29. 11:26	Fájlmappa	
 .classpath	2023. 11. 04. 13:50	CLASSPATH fájl	1 KB
 .project	2023. 11. 04. 13:48	PROJECT fájl	1 KB
 XMLHepmiu	2023. 11. 24. 22:12	XML dokumentum	3 KB
 XMLHepmiu1	2023. 12. 05. 13:25	XML dokumentum	4 KB
 XMLHepmiu2	2023. 12. 05. 13:40	XML dokumentum	5 KB