JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Hadsereg struktúra szemléltetése XML környezetben

Készítette: Urbán Olivér

Neptunkód: HEPMIU

Dátum: 2023.12.05

Tartalomjegyzék

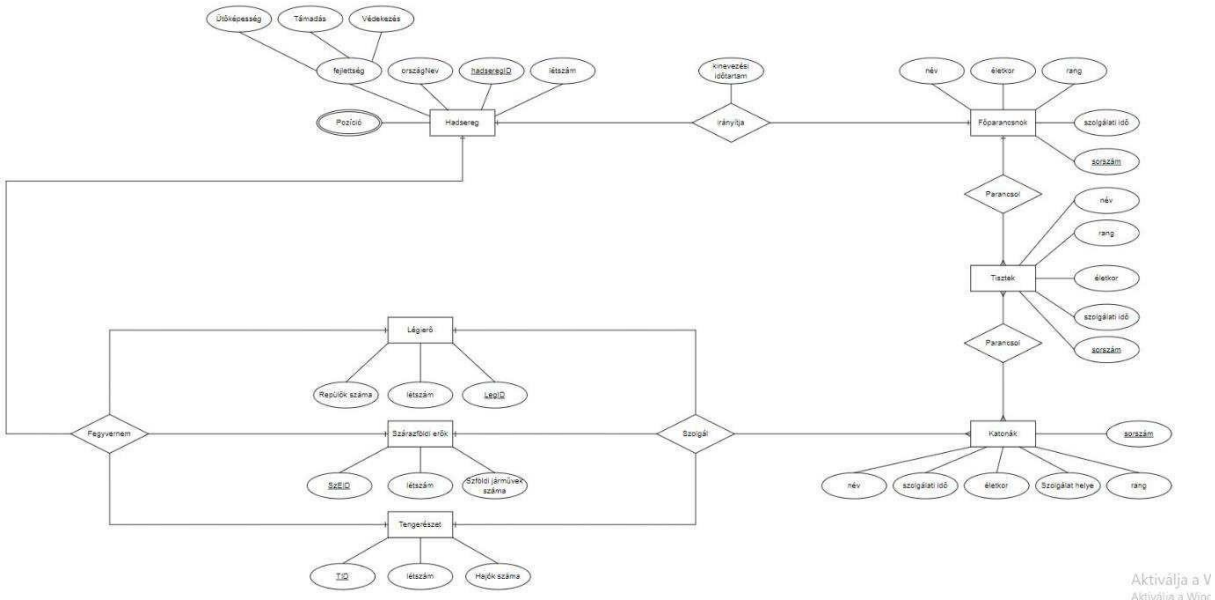
Bevezetés

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 adat lekérdező programok sikeres lefutásával bizonyosodhatunk meg, hogy XML dokumentumunk felhasználható és sikeresen beépíthető az adott környezetbe.

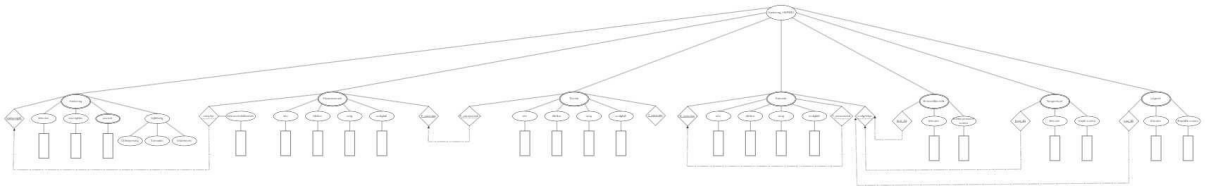
1.feladat: a) Az adatbázis ER modell tervezése

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

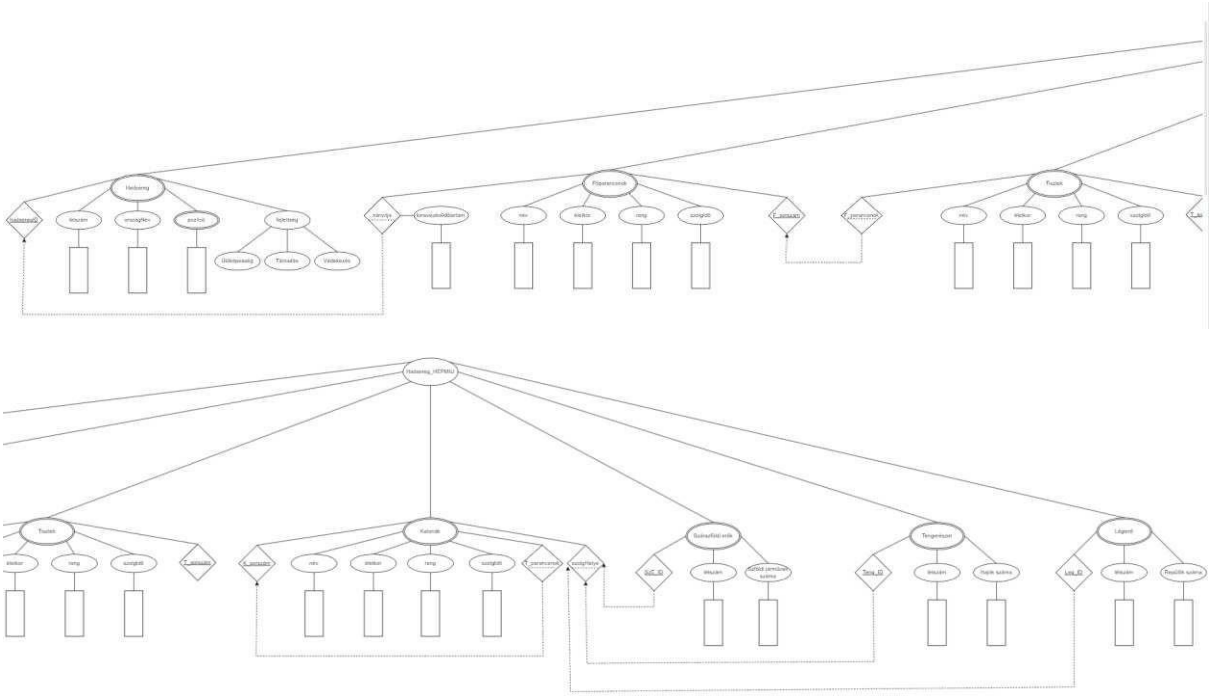


b) Adatbázis konvertálása XDM modellre

Itt látható az ER modellről átkonvertált XDM modell. Látható, hogy az ER modellben lévő egyedek egy gyökérelem gyerekei az ábrán. A tulajdonságokon kívűl 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
<?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>
```

```xml
<!--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-->
<Szárazföldi_erők SzE_ID="56">
    <létszám>5000</létszám>
    <SzföldiJárművekSzáma>250</SzföldiJárművekSzáma>

</Szárazföldi_erők>

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

<Légierő Leg_ID="18">
    <létszám>2000</létszám>
    <RepülőkSzáma>120</RepülőkSzáma>
</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_tipus">
        <xs:restriction base="xs:string">
            <xs:length value="4" />
        </xs:restriction>
    </xs:simpleType>



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


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

     <xs:simpleType name="HadnemID_tipus">
         <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="Szárazföldi_erők_kulcsa">
        <xs:selector xpath="Szárazföldi_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="Tengerészet_kulcsa">
        <xs:selector xpath="Tengerészet" />
        <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_Tengerészet_kulcsa" refer="Teng_ID">
        <xs:selector xpath="Tengerészet" />
        <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>


<!-- KOMLEX 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_tipus"
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_tipus"
use="required" />
            <xs:attribute name="F_parancsnok" type="F_sorszám_tipus"
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_tipus"
use="required" />
            <xs:attribute name="T_parancsnok" type="T_sorszám_tipus"
use="required" />
            <xs:attribute name="szolgHelye" type="HadnemID_tipus"
use="required" />
          </xs:complexType>
        </xs:element>



<!--SZFÖLDI ERŐK EGYED SÉMÁJA-->
        <xs:element name="Szárazföldi_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="Tengerészet">
          <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:

A DomReadHepmiu kód az adatolvasást végzi az XML dokumentumból az alábbi módon:

A main metódusban történik a fájl létrehozása, valamint az XMl dokumentum parse-olása a DOM által, megtörténik továbbá a megírt metódusok hívása is melynek eredményeképpen kiírja a konzolra az XML dokumentumot fa struktúrában.

A cutEmptyStrings metódus nevéhez hűen kitörli a dokumentumban található üres csomópontokat.

A writeDoc a megadott XML dokumentumot írja ki egy új fájlba.

A makeToXMLFormat véglegesíti az XML dokumentum átalakítását a fa struktúrához mérten.

Az elementsToXMLFormat adja hozzá az elementekhez, elementek gyerekeihez,attribútumaihoz a nyitó és záró tageket és visszaadja String típusként.

Az adatolvasást megvalósító kód:

```
package hu.domparse.hepmiu;

import  java.io.File;

import java.io.IOException;

import java.util.ArrayList;

import java.util.List;

import javax.xml.parsers.*;

import javax.xml.transform.OutputKeys;

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.*;

import org.xml.sax.SAXException;

public class DomReadHepmiu {

public static void main(String[] args) {

// TODO Auto-generated method stub

try {

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

StreamResult newXmlStream = new StreamResult(newXMLFile);

DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();

DocumentBuilder builder = factory.newDocumentBuilder();

Document doc = builder.parse(new File("./XMLHepmiu.xml"));

cutEmptyStrings(doc.getDocumentElement());

writeDoc(doc, newXmlStream);

System.out.println(makeToXMLFormat(doc));

} catch (Exception e) {
```

```
e.printStackTrace();
}
}
// A fájlban lévő üres stringek törlése
private static void cutEmptyStrings(Node root) {
NodeList nodeList = root.getChildNodes();
List<Node> deleteEmptyLists = new ArrayList<>();
for (int i = 0; i < nodeList.getLength(); i++) {
if (nodeList.item(i).getNodeType() == Node.TEXT_NODE
&& nodeList.item(i).getTextContent().strip().isEmpty()) {
deleteEmptyLists.add(nodeList.item(i));
} else {
cutEmptyStrings(nodeList.item(i));
}
}
for (Node node : deleteEmptyLists) {
root.removeChild(node);
}
}
// Új XML fájlba írása
public static void writeDoc(Document document, StreamResult output) {
try {
TransformerFactory transformerFactory = TransformerFactory.newInstance();
Transformer transformer = transformerFactory.newTransformer();
transformer.setOutputProperty(OutputKeys.ENCODING, "UTF-8");
transformer.setOutputProperty(OutputKeys.INDENT, "yes");
DOMSource source = new DOMSource(document);
```

```
transformer.transform(source, output);

} catch (Exception e) {

e.printStackTrace();

}

}

// XML formátum létrehozása

public static String makeToXMLFormat(Document document) {

return "<?xml version=\"" + document.getXmlVersion() + "\" encoding=\"" +
document.getXmlEncoding() + "\" ?>\n"

+ elementsToXMLFormat(document.getDocumentElement(), 0);

}

// Tag-ek rendelése az XML elementekhez

public static String elementsToXMLFormat(Node node, int indent) {

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

return "";

}

StringBuilder output = new StringBuilder();

output.append(getIndent(indent)).append("<").append(((Element)
node).getTagName());

if (node.hasAttributes()) {

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

Node attribute = node.getAttributes().item(i);

output.append("
").append(attribute.getNodeName()).append("=\"").append(attribute.getNodeValue())

.append("\"");

}

}

NodeList children = node.getChildNodes();
```

```
if (children.getLength() == 1 && children.item(0).getNodeType() == Node.TEXT_NODE)
{

output.append(">").append(children.item(0).getTextContent().trim()).append("</")

.append(((Element) node).getTagName()).append(">\n");

} else {

output.append(">\n");

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

output.append(elementsToXMLFormat(children.item(i), indent + 1));

}

output.append(getIndent(indent)).append("</").append(((Element)
node).getTagName()).append(">\n");

}

return output.toString();

}

// Space-ek beiktatása

private static String getIndent(int indent) {

StringBuilder indentation = new StringBuilder();

for (int i = 0; i < indent; i++) {

indentation.append(" ");

}

return indentation.toString();

}

}
```

```
problems Javadoc Declaration Console Coverage                                                    6
<terminated> DomReadHepmiu [Java Application] C:\Users\oliez\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_1*.0.6.v2023020d-1*\jre\bin\javaw.exe (2023. dec. 12. 10:15:
<?xml version="1.0" encoding="UTF-8" ?>
<Hadsereg HEPi'ÎlU xmlns:xs="http://\\wv.\v3.org/2661/X'îLSchena-instance" xs:noNamespaceSchenaLocation="X'îLSchenaHEPi'ÎlU.xsd">
    <Hadsereg hadseregID="1">
        <1étszän>9666</1étszän>
        <országNév>Hungary</országNév>
        <pozíció>Dé1-A1fÖ1d</pozíció>
        <fejlettség>
            <ütôképesség>kÖzepes</ütôképesség>
            <tänadäs>erôs</tänadäs>
            <védekezés>gyenge</védekezés>
        </fej1ettség>
    </Hadsereg>
    <Fôparancsnok Sorszam="4545" irányIta="1">
        <név>Nagy 3özsef</név>
        <rang>vezérezredes</rang>
        <életkor>56</életkor>
        <szo1gIdo>36</szo1gIdo>
    </Foparancsnok>
    <Tisztek Parancsnok="4545" Sorszam="2323">
        <mév>Király Zoltäm</mév>
        <rang>százados</rang>
        <élet kor>42</élet kor>
        <szo1gIdo>22</szo1gIdo>
    </Tisztek>
    <Tisztek Parancsnok="4545" Sorszam="2336">
        <név>Nagy Zso1t</név>
        <rang>ezredes</rang>
        <élet kor>48</élet kor>
        <szo1gIdo>28</szo1gIdo>
    </Tisztek>
    <Tisztek Parancsnok="4545" Sorszam="9836">
        <név>Szabö üyu1a</név>
        <rang>ornagy</rang>
        <életkor>58</életkor>
        <szo1gIdo>36</szo1gIdo>
    </Tisztek>
    <Katonäk Sorszam="7874" Parancsnok="9836" szo1gHe1ye="56">
        <név>Szabö Kristöf</név>
        <rang>tizedes</rang>
        <élet kor>26</élet kor>
        <szo1gIdo>1</szo1gIdo>
    </Katonäk>
    <Katonäk Sorszam="4921" Parancsnok="9836" szo1gHe1ye="56">
        <név>Töth Bendeguz</név>
        <rang>hadnagy</rang>
        <élet kor>28</élet kor>
        <szo1gIdo>16</szo1gIdo>
    </Katonäk>
    <Katonäk Sorszam="7123" Parancsnok="9836" szo1gHe1ye="56">
        <név>Töth Bendeguz</név>
        <rang>tizedes</rang>
        <élet kor>23</élet kor>
```

2b) adatmódosítás:

A main metódusban a fájl létrehozás és parse-olás után a felhasználó módosíthatja az XML fájl elementjeit, ezt egy egyszerű Scanner implementálásával végeztem. A usernek meg kell adnia az element nevét, ID-ját, Node nevét, attributumát és az új értéket. A többit a meghívott metódusok végzik el.

A modifyElementByID végzi a módosítást az inputba megadott értékek alapján. Először megkeresi a megfelelő névvel és Id-val rendelkező Node-ot. Amint ez megtörtént, hogy az inputban megadott, változtatni kívánt element nevét megtalálta, akkor az új értéket beállítja.

A wrtiteToFile metódus az elvégzett módosításokat egy új fájlba írja ki.

Valamint a konzolra kiírunk egy üzenetet, ha a módosítás sikeres volt.

Íme az adatmódosítást megvalósító kód:

```
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("Szárazföldi_erők") ||
elementName.equalsIgnoreCase("Tengerészet")

|| 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 {
```

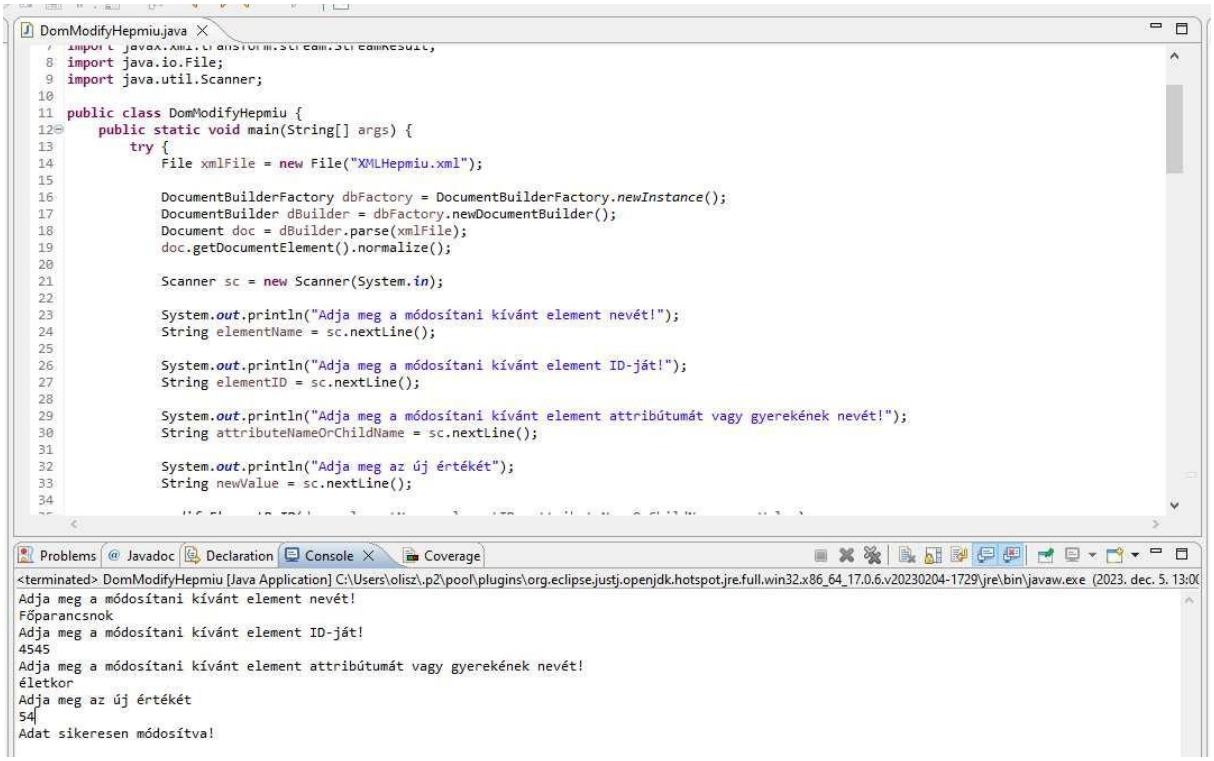
```
NodeList childNodes = element.getElementsByTagName(attributeNameOrChildName);
if (childNodes.getLength() > 0) {
Node childNode = childNodes.item(0);
childNode.setTextContent(newValue);
} else {
System.out.println("Adat típus nem található: " + attributeNameOrChildName);
}
}
}
}
else if (elementName.equalsIgnoreCase("Hadsereg"))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

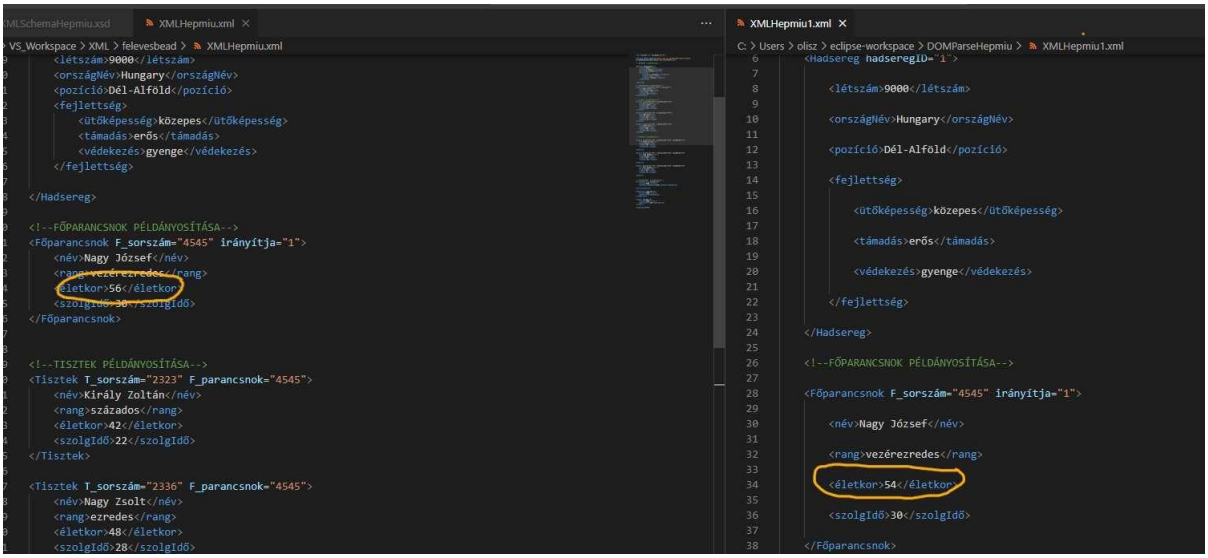
```
}

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 következő képernyőképeken az elvégzett módosítások láthatóak. Először a fejlesztőkörnyezetben végrehajtott input sorozatot láthatjuk, majd a fájlok összehasonlítását, eredmény ellenőrzését látjuk kiemelve.

A módosítandó elemeket a képernyőfotók előtt tisztáztam.

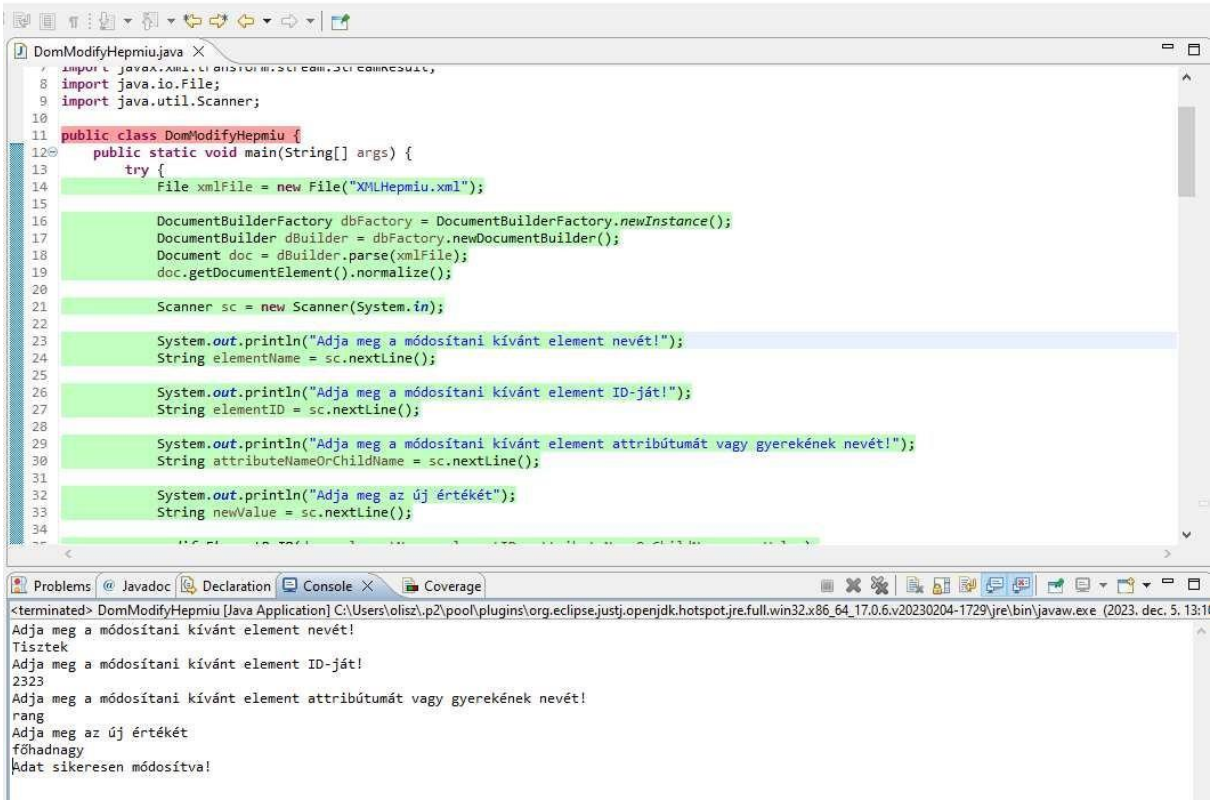
A főparancsnok életkorának módosítása:





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

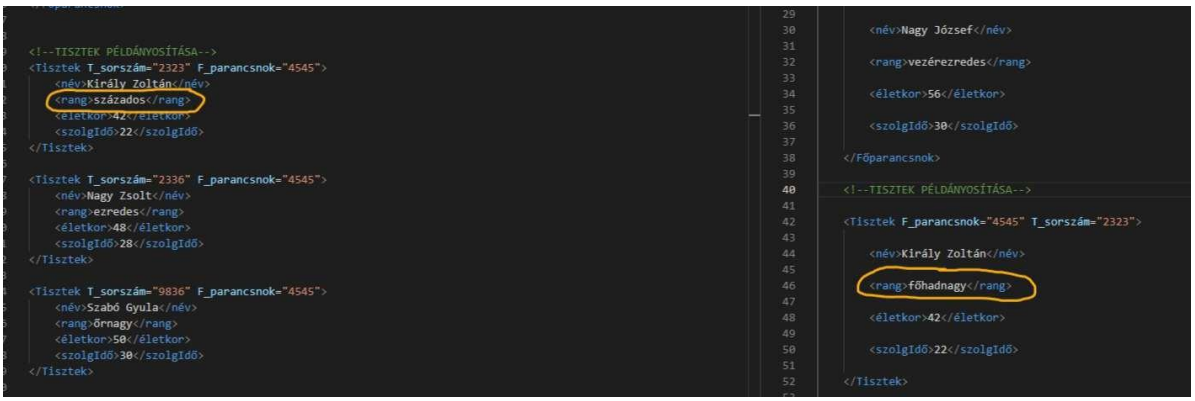
Eclipse IDE



```java
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();
```

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. 13:1
Adja meg a módosítani kívánt element nevét!
Tisztek
Adja meg a módosítani kívánt element ID-ját!
2323
Adja meg a módosítani kívánt element attribútumát vagy gyerekének nevét!
rang
Adja meg az új értékét
főhadnagy
Adat sikeresen módosítva!
```

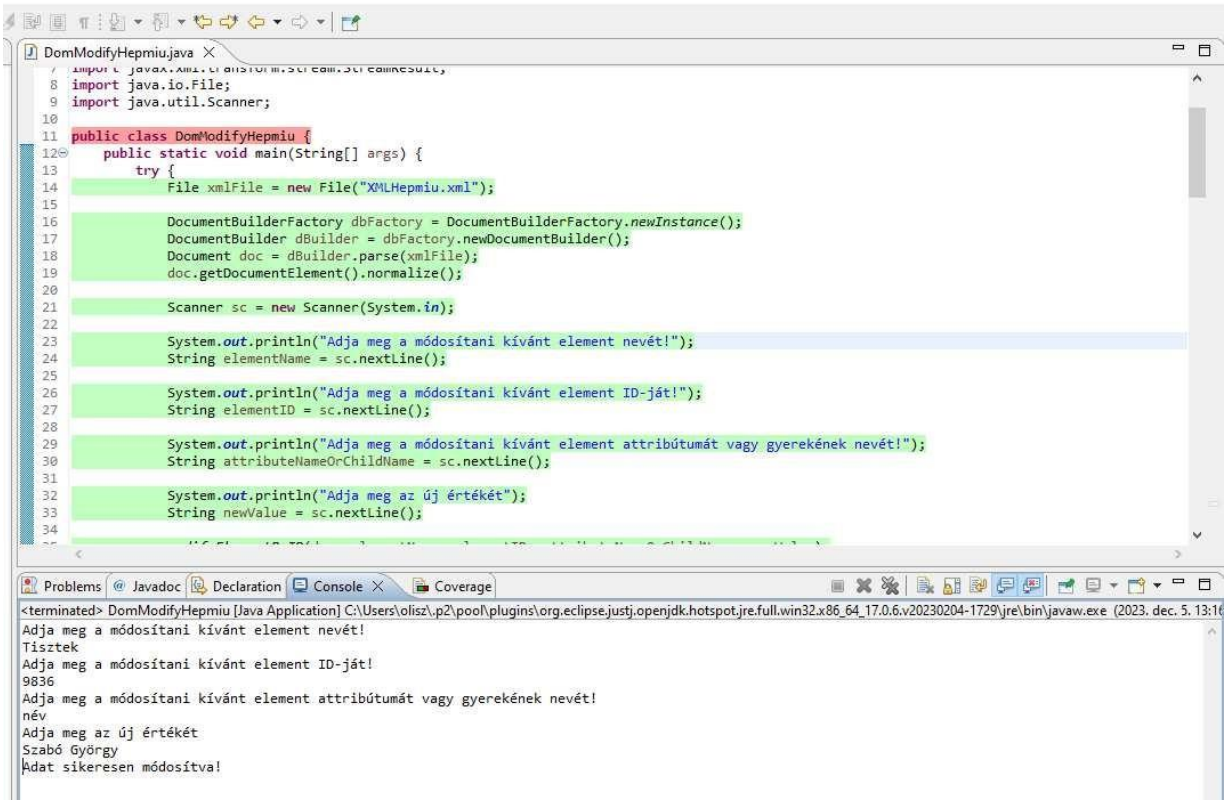


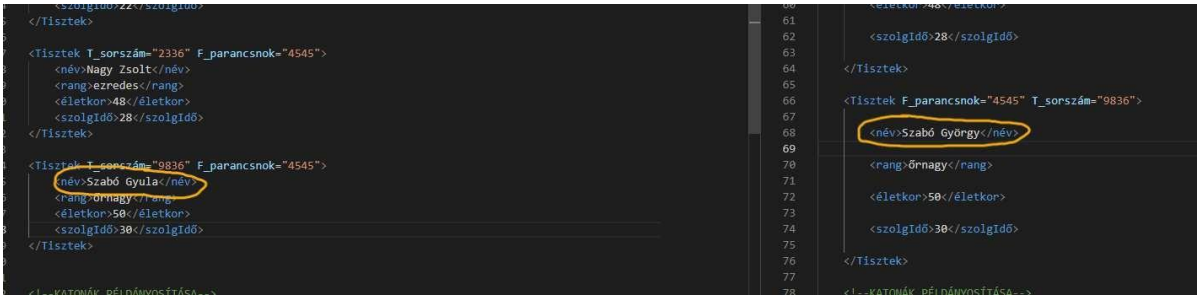
```
<!--TISZTEK PÉLDÁNYOSÍTÁSA-->
<Tisztek T_sorszám="2323" F_parancsnok="4545">
    <név>Király Zoltán</név>
    <rang>százados</rang>
    <életkor>42</életkor>
    <szolgIdő>22</szolgIdő>
</Tisztek>

<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>
```

```
        <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





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

:clipse IDE

DomModifyHepmiu.java

```
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();
```

Problems | Javadoc | Declaration | Console | Coverage

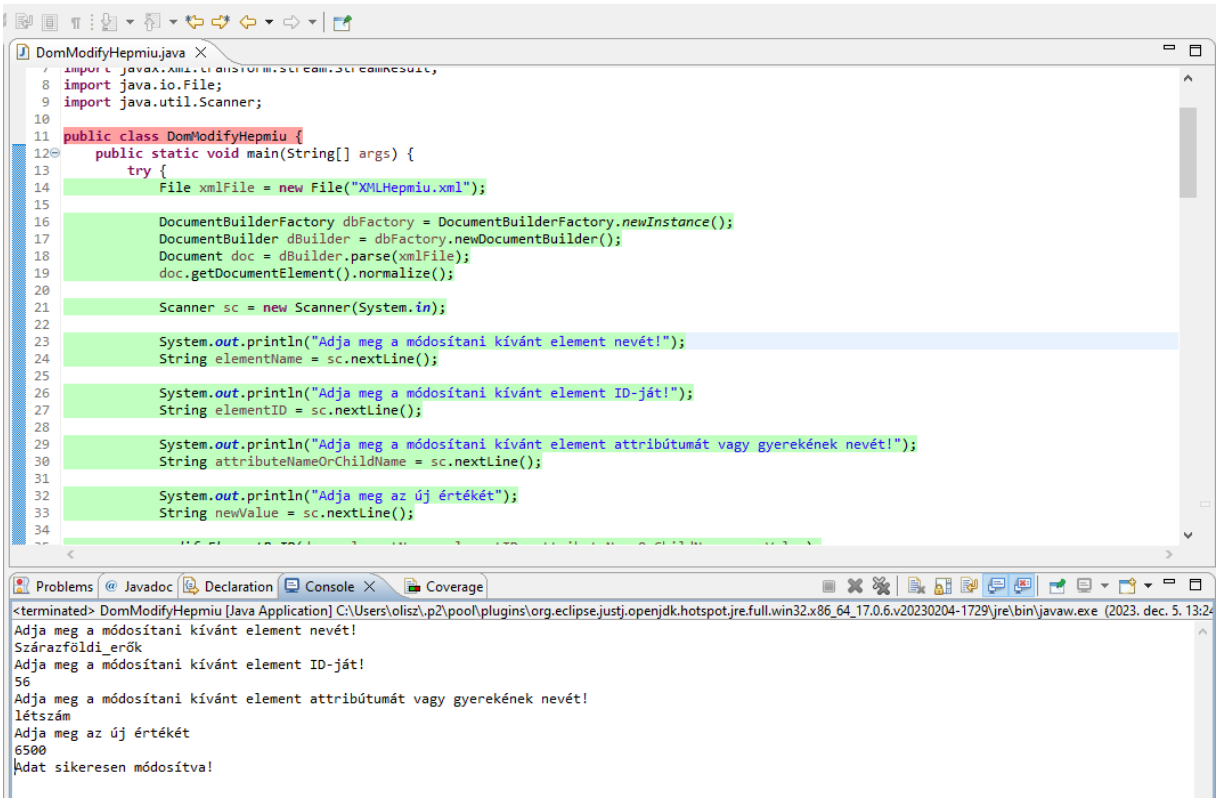
```
<terminated> DomModifyHepmiu [Java Application] C:\Users\olisz\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204-1729\jre\bin\javaw.exe (2023. dec. 5. 13:
Adja meg a módosítani kívánt element nevét!
Katonák
Adja meg a módosítani kívánt element ID-ját!
4921
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>
```

```
        <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>6</szolgIdő>

    </Katonák>
```

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

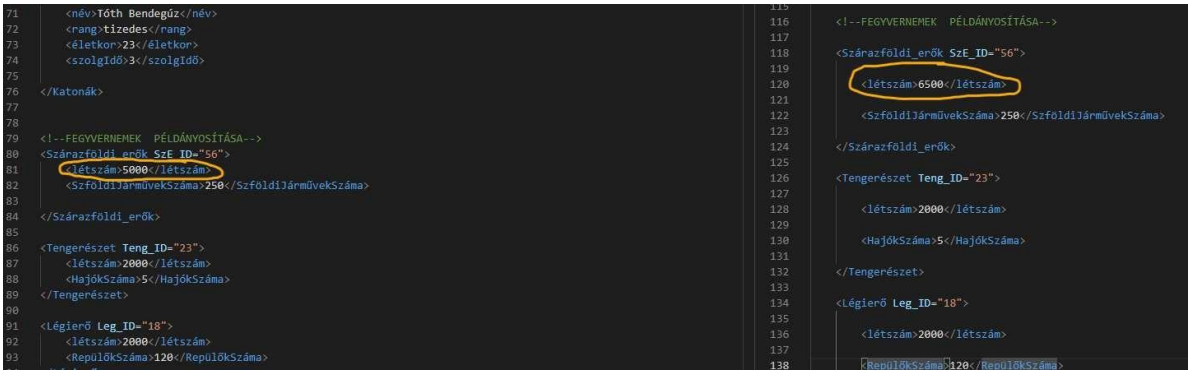
Eclipse IDE



```java
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();
```

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. 13:24
Adja meg a módosítani kívánt element nevét!
Szárazföldi_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!
```



```
    <név>Tóth Bendegúz</név>
    <rang>tizedes</rang>
    <életkor>23</életkor>
    <szolgIdő>3</szolgIdő>

</Katonák>


<!--FEGYVERNEMEK  PÉLDÁNYOSÍTÁSA-->
<Szárazföldi_erők SzE_ID="56">
    <létszám>5000</létszám>
    <SzföldiJárművekSzáma>250</SzföldiJárművekSzáma>

</Szárazföldi_erők>

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

<Légierő Leg_ID="18">
    <létszám>2000</létszám>
    <RepülőkSzáma>120</RepülőkSzáma>
```

```
<!--FEGYVERNEMEK  PÉLDÁNYOSÍTÁSA-->

<Szárazföldi_erők SzE_ID="56">

    <létszám>6500</létszám>

    <SzföldiJárművekSzáma>250</SzföldiJárművekSzáma>

</Szárazföldi_erők>

<Tengerészet Teng_ID="23">

    <létszám>2000</létszám>

    <HajókSzáma>5</HajókSzáma>

</Tengerészet>

<Légierő Leg_ID="18">

    <létszám>2000</létszám>

    <RepülőkSzáma>120</RepülőkSzáma>
```

2c) adat lekérdezés:

Az adat lekérdezést megvalósító program ugyancsak a fájl létrehozásával és parsolásával kezdődik a main-ben. Ezután a módosítást végző kódomhoz hasonlóan a felhasználó inputok megadásával kérdezheti le az elementeket. Itt elég az element nevét és ID-ját megadni. Ezt követően hívódik meg a selectElementByID metódus, amely végig nézi a fájlt a megadot inputok alapján. Ugyanúgy végig iterálja a Node-okat, Node-ok gyerekeit. Ha megtalálta vissza adja az elementhez tartozó összes adatot, értékeikkel együtt.

Az adat lekérdezést megvalósító kód:

```
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();
```

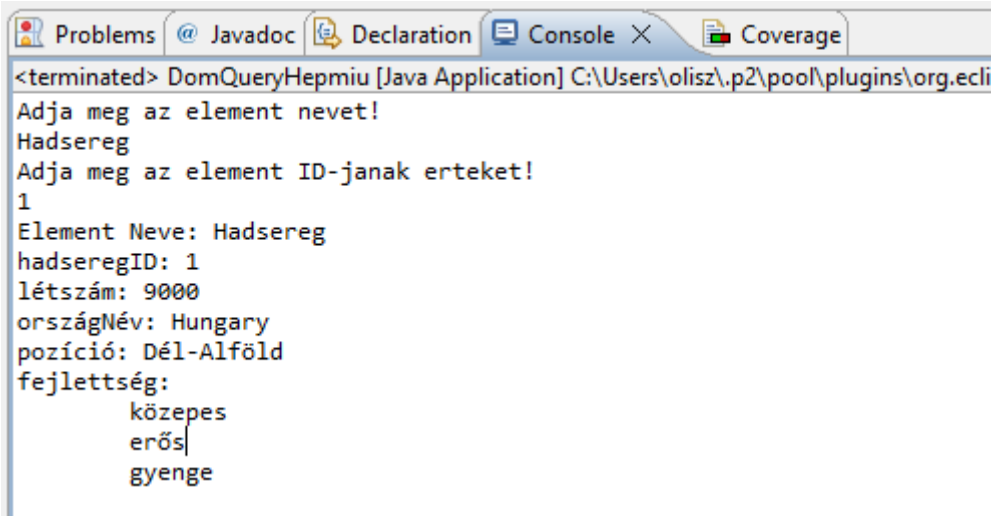
```
System.out.println("Adja meg az element ID-janak erteket!");

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("Tengerészet")

|| 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());

}

}

}
```

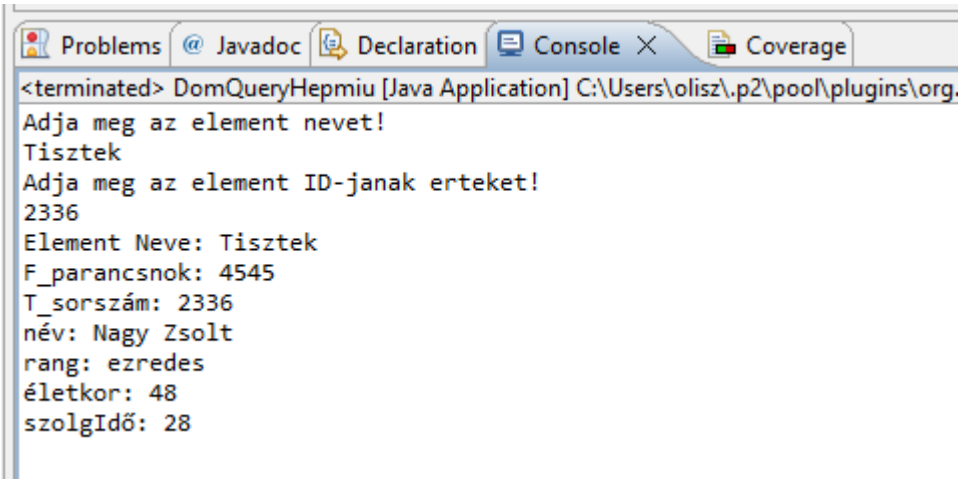
```
}
else if (elementName.equalsIgnoreCase("Hadsereg")) {
if (element.getAttribute("hadseregID").equals(elementID)) {
// ELEMENTEK KIÍRÁSA ÉRTÉKEIKKEL EGYÜTT
System.out.println("Element Neve: " + element.getNodeName());
NamedNodeMap attributes = element.getAttributes();
for (int j = 0; j < attributes.getLength(); 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());
}
}
}
}
}
}
}
}
}
}
}
}
```

}

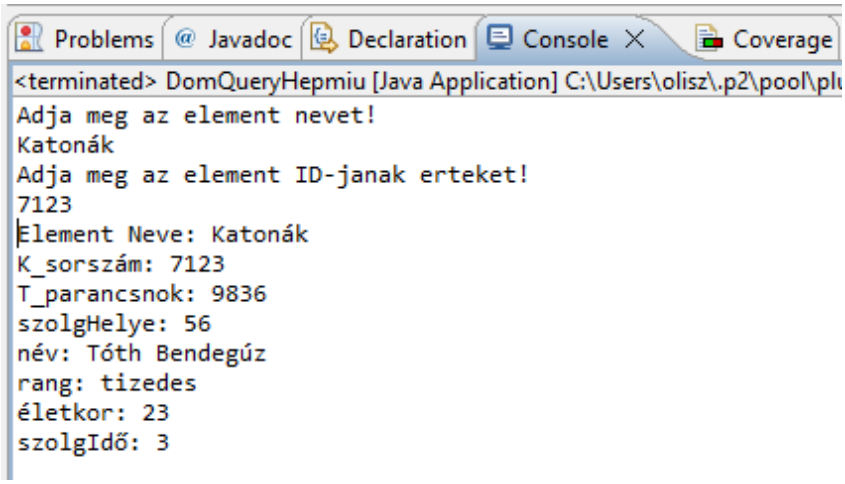
Hadsereg element lekérdezése



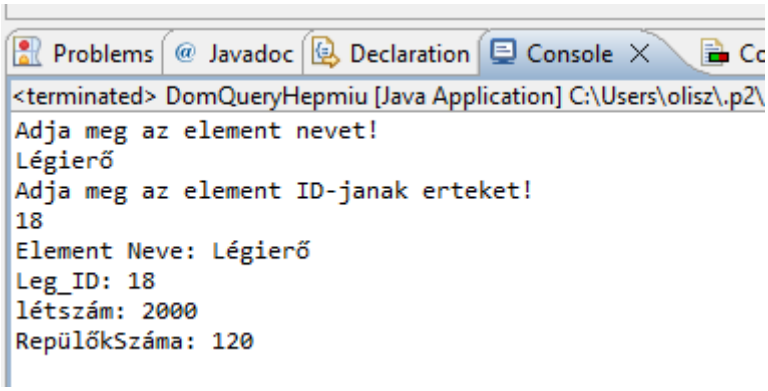
„2336” sorszámú Tiszt adatainak lekérdezése



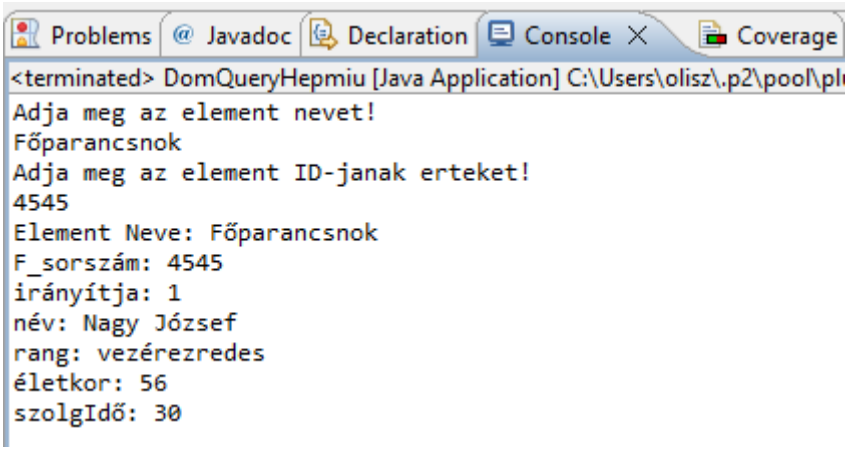
„7123” sorszámú Katona lekérdezése



Légierő element adatainak lekérdezése



Főparancsnok adatainak lekérdezése



2d) adatírás:

A writeElementToFileAndConsole metódus hívásával indul a programunk, ez előkészíti a dokumentumot, majd létrehozza a gyökérelemet és hozzáad elemeket a dokumentumhoz az addElemets metódus segítségével.

Az addElements metódus hozzáad különböző típusú katonai egységek adatait a dokumentumhoz, az `addHadsereg`, `addFoparancsnok`, `addTisztek`, `addKatonak`, `addSzarazfoldierok`, `addTengereszet`, `addLegiero` metódusok segítségével. Ezek a metódusokban történik egy element struktúrájának, gyerekeinek és attribútumainak meghatározása.

A printDocument kiírja a dokumentumot a konzolra és egy XML fájlba. A kimeneti formázást és indentálást is kezeli. A saveDocument elvégzi a dokumentum mentéséhez szükséges műveleteket, például beállítja a kimeneti tulajdonságokat és meghívja a printDocument metódust.

Végül egy XML fájlt generál a XMLHepmiu1.xml néven.

```
package hu.domparse.hepmiu;

import  javax.xml.parsers.DocumentBuilder;

import javax.xml.parsers.DocumentBuilderFactory;

import javax.xml.transform.OutputKeys;

import  javax.xml.transform.Transformer;

import javax.xml.transform.TransformerFactory;

import java.io.File;

import java.io.FileWriter;

import java.io.PrintWriter;

import java.util.Arrays;

import java.util.List;

import java.util.StringJoiner;

import org.w3c.dom.*;

public class DOMWriteHepmiu {

public static void main(String[] args) {

writeElementsToFileAndConsole();

}

private static void writeElementsToFileAndConsole() {

try {

Document document = prepareDocument();

Element rootElement = document.createElement("Hadsereg_HEPMIU");

document.appendChild(rootElement);
```

```
addElements(document, rootElement);

saveDocument(document);

} catch (Exception e) {

e.printStackTrace();

}

}

private static Document prepareDocument() throws Exception {

DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();

DocumentBuilder builder = factory.newDocumentBuilder();

Document document = builder.newDocument();

return document;

}

private static void addElements(Document document, Element rootElement) {

addHadsereg(document, rootElement,"1", "9000",

"Hungary","Dél-Alföld","közepes", "erős", "gyenge");

addFoparancsnok(document, rootElement, "4545", "1", "Nagy József", "vezérezredes",
"56", "30");

addTisztek(document, rootElement, "2323", "4545", "Király Zoltán", "százados",
"42", "22");

addTisztek(document, rootElement, "2336", "4545", "Nagy Zsolt", "ezredes", "48",
"28");

addTisztek(document, rootElement, "9836", "4545", "Szabó Gyula", "őrnagy", "50",
"30");

addKatonak(document, rootElement, "7874", "9836", "56", "Szabó Kristóf",
"tizedes", "20", "1");

addKatonak(document, rootElement, "4921", "9836", "56", "Tóth Bendegúz",
"hadnagy", "28", "10");

addKatonak(document, rootElement, "7123", "9836", "56", "Kovács István",
"tizedes", "23", "3");

addSzarazfoldierok(document, rootElement, "56", "5000", "250");

addTengereszet(document, rootElement, "23", "2000", "5");
```

```
addLegiero(document, rootElement, "18", "2000", "120");

}

private static void saveDocument(Document document) {

try {

TransformerFactory transformerFactory = TransformerFactory.newInstance();

Transformer transformer = transformerFactory.newTransformer();

transformer.setOutputProperty(OutputKeys.ENCODING, "UTF-8");

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

transformer.setOutputProperty("{https://xml.apache.org/xslt}indent-amount", "4");

printDocument(document);

} catch (Exception e) {

e.printStackTrace();

}

}

private static void printDocument(Document document) {

try {

File xmlFile = new File("XMLHepmiu1.xml");

PrintWriter writer = new PrintWriter(new FileWriter(xmlFile, true));

Element rootElement = document.getDocumentElement();

String rootName = rootElement.getTagName();

StringJoiner rootAttributes = new StringJoiner(" ");

NamedNodeMap rootAttributeMap = rootElement.getAttributes();

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

Node attribute = rootAttributeMap.item(i);

rootAttributes.add(attribute.getNodeName() + "=\"" + attribute.getNodeValue() +
"\"");

}

System.out.print("<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n");
```

```
writer.print("<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n");

System.out.print("<" + rootName + " " + rootAttributes.toString() + ">\n");

writer.print("<" + rootName + " " + rootAttributes.toString() + ">\n");

System.out.println("</" + rootName + ">");

writer.append("</" + rootName + ">");

writer.close();

} catch (Exception e) {

e.printStackTrace();

}

}

private static void addHadsereg(Document document, Element rootElement, String
hadseregID, String létszam,

String orszagNev, String pozicio, String utoképessseg, String tamadas, String
vedelekezes) {

Element hadsereg = document.createElement("Hadsereg");

hadsereg.setAttribute("hadseregID", hadseregID);

Element letszamElement = createElement(document, "létszám", létszam);

hadsereg.appendChild(letszamElement);

Element orszagNevElement = createElement(document, "országNév", orszagNev);

hadsereg.appendChild(orszagNevElement);

Element pozicioElement = createElement(document, "pozíció", pozicio);

hadsereg.appendChild(pozicioElement);

Element fejlettségElement = document.createElement("fejlettség");

Element utoképességElement = createElement(document, "ütőképesség", utoképessseg);
```

```
fejlettségElement.appendChild(utoképességElement);

Element tamadasElement = createElement(document, "támadás", tamadas);

fejlettségElement.appendChild(tamadasElement);

Element vedelekezesElement = createElement(document, "védekezés", vedelekezes);

fejlettségElement.appendChild(vedelekezesElement);

hadsereg.appendChild(fejlettségElement);

rootElement.appendChild(hadsereg);

}

private static void addFoparancsnok(Document document, Element rootElement, String
fsorszam, String iranyitja,

String nev, String rang, String eletkor, String szolgIdo) {

Element foparancsnok = document.createElement("Főparancsnok");

foparancsnok.setAttribute("F_sorszám", fsorszam);

foparancsnok.setAttribute("irányítja", iranyitja);

Element nevElement = createElement(document, "név", nev);

Element rangElement = createElement(document, "rang", rang);

Element eletkorElement = createElement(document, "életkor", eletkor);

Element szolgIdoElement = createElement(document, "szolgIdő", szolgIdo);

foparancsnok.appendChild(nevElement);

foparancsnok.appendChild(rangElement);

foparancsnok.appendChild(eletkorElement);

foparancsnok.appendChild(szolgIdoElement);

rootElement.appendChild(foparancsnok);

}

private static void addTisztek(Document document, Element rootElement, String
tsorszam, String fparancsnok,

String nev, String rang, String eletkor, String szolgIdo) {

Element tisztek = document.createElement("Tisztek");
```

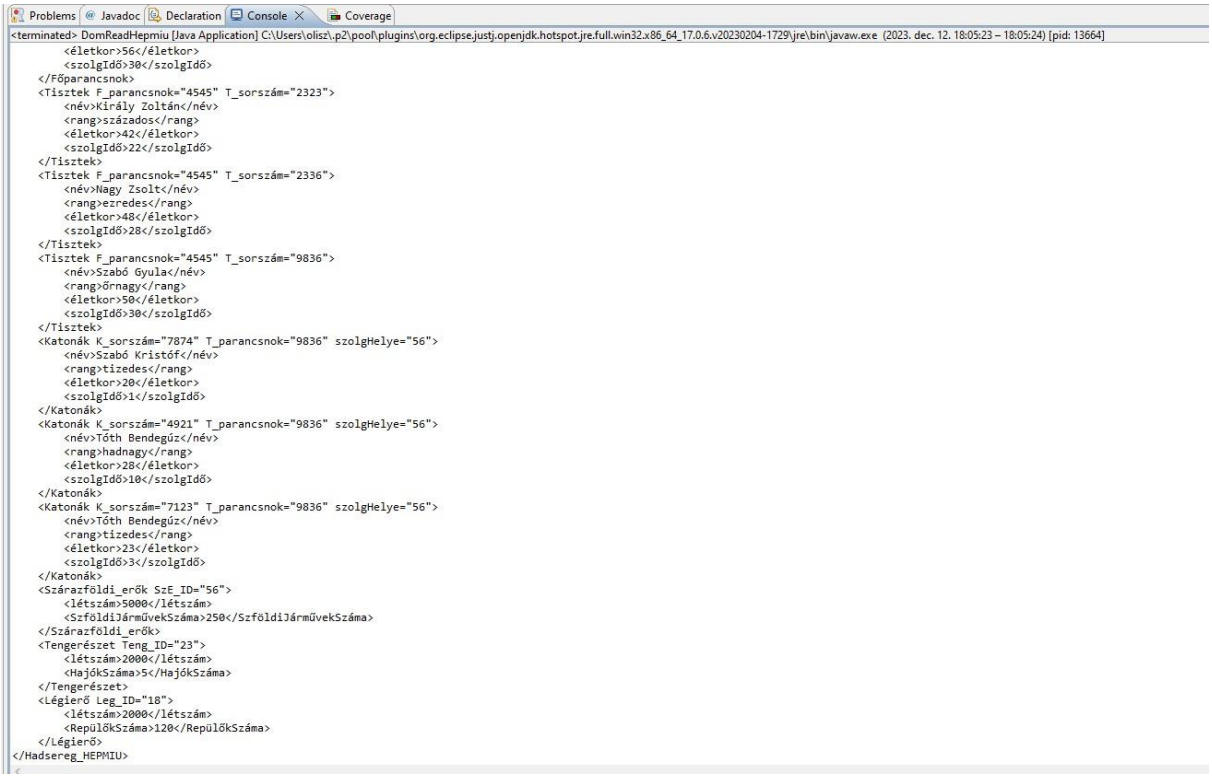
```
tisztek.setAttribute("T_sorszám", tsorszam);

tisztek.setAttribute("F_parancsnok", fparancsnok);

Element nevElement = createElement(document, "név", nev);

Element rangElement = createElement(document, "rang", rang);

Element eletkorElement = createElement(document, "életkor", eletkor);

Element szolgIdoElement = createElement(document, "szolgIdő", szolgIdo);

tisztek.appendChild(nevElement);

tisztek.appendChild(rangElement);

tisztek.appendChild(eletkorElement);

tisztek.appendChild(szolgIdoElement);

rootElement.appendChild(tisztek);

}

private static void addKatonak(Document document, Element rootElement, String
ksorszam, String tparancsnok,

String szolgHelye, String nev, String rang, String eletkor, String szolgIdo) {

Element katonak = document.createElement("Katonák");

katonak.setAttribute("K_sorszám", ksorszam);

katonak.setAttribute("T_parancsnok", tparancsnok);

katonak.setAttribute("szolgHelye", szolgHelye);

Element nevElement = createElement(document, "név", nev);

Element rangElement = createElement(document, "rang", rang);

Element eletkorElement = createElement(document, "életkor", eletkor);

Element szolgIdoElement = createElement(document, "szolgIdő", szolgIdo);

katonak.appendChild(nevElement);

katonak.appendChild(rangElement);

katonak.appendChild(eletkorElement);

katonak.appendChild(szolgIdoElement);

rootElement.appendChild(katonak);
```

```
}

private static void addSzarazfoldierok(Document document, Element rootElement,
String szE_ID, String letszam,

String szfoldiJarmuvekSzama) {

Element szarazfoldierok = document.createElement("Szárazföldi_erők");

szarazfoldierok.setAttribute("SzE_ID", szE_ID);

Element letszamElement = createElement(document, "létszám", letszam);

Element szfoldiJarmuvekSzamaElement = createElement(document,
"SzföldiJárművekSzáma", szfoldiJarmuvekSzama);

szarazfoldierok.appendChild(letszamElement);

szarazfoldierok.appendChild(szfoldiJarmuvekSzamaElement);

rootElement.appendChild(szarazfoldierok);

}

private static void addTengereszet(Document document, Element rootElement, String
teng_ID, String letszam,

String hajokSzama) {

Element tengereszet = document.createElement("Tengerészet");

tengereszet.setAttribute("Teng_ID", teng_ID);

Element letszamElement = createElement(document, "létszám", letszam);

Element hajokSzamaElement = createElement(document, "HajókSzáma", hajokSzama);

tengereszet.appendChild(letszamElement);

tengereszet.appendChild(hajokSzamaElement);

rootElement.appendChild(tengereszet);

}

private static void addLegiero(Document document, Element rootElement, String
leg_ID, String letszam,

String repulokSzama) {

Element legiero = document.createElement("Légierő");

legiero.setAttribute("Leg_ID", leg_ID);
```

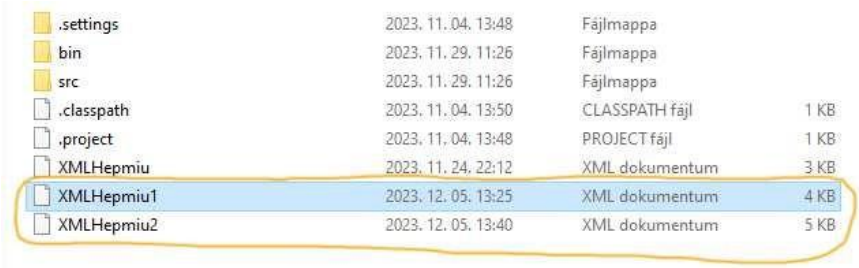
```
Element letszamElement = createElement(document, "létszám", letszam);

Element repulokSzamaElement = createElement(document, "RepülőkSzáma",
repulokSzama);

legiero.appendChild(letszamElement);

legiero.appendChild(repulokSzamaElement);

rootElement.appendChild(legiero);

}

private static Element createElement(Document document, String name, String value)
{

Element element = document.createElement(name);

element.appendChild(document.createTextNode(value));

return element;

}

private static void printNodeList(NodeList nodeList, PrintWriter writer) {

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

Node node = nodeList.item(i);

printNode(node, 1, writer);

System.out.println("");

writer.println("");

}

}

private static void printNode(Node node, int indent, PrintWriter writer) {

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

Element element = (Element) node;

String nodeName = element.getTagName();

StringJoiner attributes = new StringJoiner(" ");

NamedNodeMap attributeMap = element.getAttributes();

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

```
Node attribute = attributeMap.item(i);

attributes.add(attribute.getNodeName() + "=\"" + attribute.getNodeValue() + "\"");

}

System.out.print(getIndentString(indent));

System.out.print("<" + nodeName + " " + attributes.toString() + ">");

writer.print(getIndentString(indent));

writer.print("<" + nodeName + " " + attributes.toString() + ">");

NodeList children = element.getChildNodes();

if (children.getLength() == 1 && children.item(0).getNodeType() == Node.TEXT_NODE)
{

System.out.print(children.item(0).getNodeValue());

writer.print(children.item(0).getNodeValue());

} else {

System.out.println();

writer.println();

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

printNode(children.item(i), indent + 1, writer);

}

System.out.print(getIndentString(indent));

writer.print(getIndentString(indent));

}

System.out.println("</" + nodeName + ">");

writer.println("</" + nodeName + ">");

}

}

private static String getIndentString(int indent) {

StringBuilder sb = new StringBuilder();

for (int i = 0; i < indent; i++) {
```

```
sb.append(" ");

}

return sb.toString();

}

}
```



```
<terminated> DomReadHepmiu [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. 12. 18:05:23 – 18:05:24) [pid: 13664]
        <életkor>56</életkor>
        <szolgIdő>30</szolgIdő>
    </Főparancsnok>
    <Tisztek F_parancsnok="4545" T_sorszám="2323">
        <név>Király Zoltán</név>
        <rang>százados</rang>
        <életkor>42</életkor>
        <szolgIdő>22</szolgIdő>
    </Tisztek>
    <Tisztek F_parancsnok="4545" T_sorszám="2336">
        <név>Nagy Zsolt</név>
        <rang>ezredes</rang>
        <életkor>48</életkor>
        <szolgIdő>28</szolgIdő>
    </Tisztek>
    <Tisztek F_parancsnok="4545" T_sorszám="9836">
        <név>Szabó Gyula</név>
        <rang>őrnagy</rang>
        <életkor>50</életkor>
        <szolgIdő>30</szolgIdő>
    </Tisztek>
    <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>
    <Szárazföldi_erők SzE_ID="56">
        <létszám>5000</létszám>
        <SzföldiJárművekSzáma>250</SzföldiJárművekSzáma>
    </Szárazföldi_erők>
    <Tengerészet Teng_ID="23">
        <létszám>2000</létszám>
        <HajókSzáma>5</HajókSzáma>
    </Tengerészet>
    <Légierő Leg_ID="18">
        <létszám>2000</létszám>
        <RepülőkSzáma>120</RepülőkSzáma>
    </Légierő>
</Hadsereg_HEPMIU>
```



.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