

# Jegyzőkönyv

Adatkezelés XML környezetben  
Féléves feladat

Készítette: **Kaló Dániel**  
Neptunkód: **BBKHXF**

## **A feladat ismertetése:**

Feladatomnak, az egyeztetés után egy autócentrum felépítését választottam, mivel úgy gondoltam, hogy ez a feladat megfelelő kihívást jelent a számomra.

Az autócentrum részei:

- **Autók**, amelyhez az alábbi részletek tartotnak:
  - Ár
  - Szín
  - Hajtás
  - Extrák
  - Motor
    - Hengerűrtartalom
    - Teljesítmény
- **Vásárlás**, amely leírja egy esetleges vásárlás részleteit, mint például
  - Darabszám
  - Fizetési mód
- **Megrendelő**
  - Név
  - Cím
    - Irányítószám
    - Település
    - Utca
    - Házszám
- **Beszállító**
  - Elérhetőség
  - Márkanév
  - Cím
    - Irányítószám
    - Település
    - Utca
    - Házszám
- **Kereskedés**
  - Autó jellemzői
  - Tartalmaz (megtalálható-e az adott autó a kereskedésben?)
- **Dolgozó**
  - Beosztás
  - Fizetés
  - Cím
    - Irányítószám
    - Település
    - Utca
    - Házszám
- **Számla**
  - Dátum
  - Összeg

Ezek az entitások tartalmaznak egyszerű, összetett, többértékű valamint származtatott tulajdonságokat is.

Pl:

Ár -> egyszerű  
Hajtás->Többértékű  
Motor->Összetett  
Kor->Származtatott

Összesen 7 entitás található a modellben

Kapcsolatokból is több féle található meg:

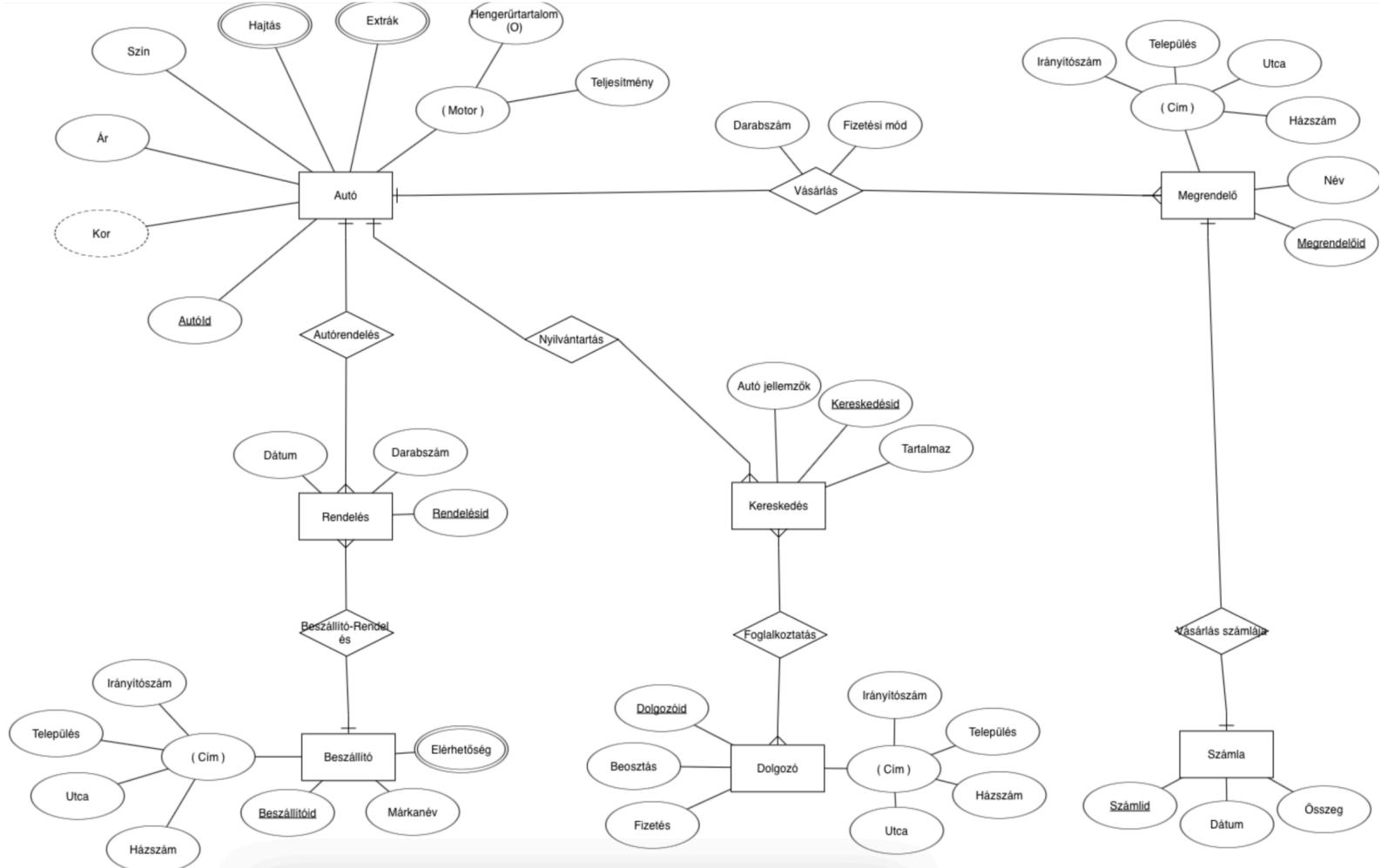
Pl:

Az autó és a megrendelő között 1:N kapcsolat van

A dolgozó és a kereskedés között N:M kapcsolat  
 Valamint a megrendelő és a számla között 1:1 kapcsolat áll fent.

1. Feladat:

## Az adatbázis ER modell:



### Rövid ismertető:

Az **autó** egyed rendelkezik egy összetett „motor” tulajdonsággal, melyhez teljesítmény mindenéppen tartozik, valamint opcionálisan „hengerürtartalom” nem elektromos autó esetén. Van többértékű „extrák” és „hajtás” tulajdonság, hiszen mind hajtásból, mind extrákból többféle lehet, ezek mellett „szín”, „ár”, „évjárat”, egyedi autó azonosító. Tartalmaz még egy kor származtatott tulajdonságot.

Van maga a **kereskedés**, mely rendelkezik összefoglalóan az autók jellemzőivel, a kereskedés azonosítójával és hogy tartalmazza-e jelenleg az autót vagy sem.

A következő a **megrendelő**, akinek van egy összetett típusú „címe”, mely az „irányítószámból”, a „településből”, az „utcából” és a „házzámból” épül fel. Tartozik még hozzá egy „név”, valamint egy egyedi azonosító.

A **dolgozó** táblához tartozik egy többértékű beosztás tulajdonság, ugyanis egy dolgozó több területen is dolgozhat felváltva. E mellett van a fizetés, valamint egy összetett cím tulajdonság.

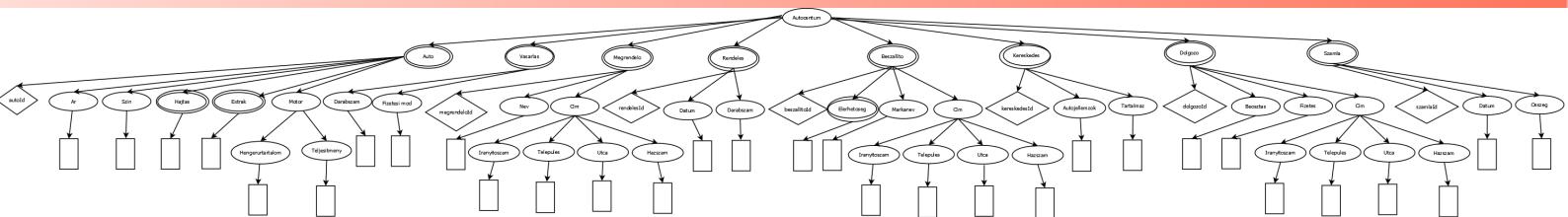
Minden vásárláshoz tartozik egy **számla**, mely az „összeget”, a „dátumot” és magát a számla azonosítót foglalja magában.

A készlet fenntartása érdekében rendelni is kell autókat, melyhez a **Rendelés** és a **Beszállító** elem tartozik.

A rendeléshez tartozik egy egyedi azonosító, egy „dátum”, valamint egy „darabszám”.

A beszállítónak szintén van egyedi azonosítója, egy „márkaneve”, valamint egy összetett „cím” tulajdonsága, mely az „irányítószámból”, a „településből”, az „utcából” és a „házzámból” épül fel.

## **ER modell konvertálása XDM modellre:**



Dupla bekarikázással jelöltem a többértékű tulajdonságokat, valamint igyekeztem érzékeltetni, ahogy vesszük sorra a gyerekelemeket, úgy haladunk egyre mélyebbre a fában.

## **Az XML elkészítése az XDM alapján:**

```
<autocentrum xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="XMLSchemaBBKHXF.xsd">
<auto autoId="1">
<ar>12000000</ar>
<szin>Fekete</szin>
<hajtas>Osszkerek</hajtas>
<extrak>Ulesfutes</extrak>
<extrak>Savtarto</extrak>
<motor>
<hengerurtartalom>1300</hengerurtartalom>
<teljesitmeny>100</teljesitmeny>
</motor>
</auto>
<auto autoId="2">
<ar>16000000</ar>
<szin>Feher</szin>
<hajtas>Osszkerek</hajtas>
<hajtas>Hatsokerek</hajtas>
<extrak>Ulesfutes</extrak>
<extrak>Savtarto</extrak>
<extrak>Kormenyfutes</extrak>
<motor>
<hengerurtartalom>1600</hengerurtartalom>
<teljesitmeny>160</teljesitmeny>
</motor>
</auto>
<vasarlas>
<darabszam>1</darabszam>
<fizetesi>Kartyas</fizetesi>
</vasarlas>
```

```
<vasarlas>
<darabszam>1</darabszam>
<fizetesimod>Keszpenz</fizetesimod>
</vasarlas>
<megrendelo megrendeloid="1">
<nev>Kiss Ákos</nev>
<cim>
<iranyitoszam>3400</iranyitoszam>
<telepules>Mezokovesd</telepules>
<utca>Szomolyai</utca>
<hazszam>12</hazszam>
</cim>
</megrendelo>
<rendeles rendelesId="1">
<datum>2021-02-04</datum>
<darabszam>5</darabszam>
</rendeles>
<beszallito beszallitoId="1">
<elerhetoseg>0620/113-5264</elerhetoseg>
<elerhetoseg>beszallito@gmail.com</elerhetoseg>
<markanev>Ferrari</markanev>
<cim>
<iranyitoszam>3500</iranyitoszam>
<telepules>Miskolc</telepules>
<utca>Tapolcai</utca>
<hazszam>45</hazszam>
</cim>
</beszallito>
<kereskedes kereskedesId="1">
<autojellemzok>Kombi</autojellemzok>
<tartalmaz>1</tartalmaz>
</kereskedes>
<dolgozo dolgozoId="1">
<beosztas>Uzletvezeto</beosztas>
<fizetes>400000</fizetes>
<cim>
<iranyitoszam>3400</iranyitoszam>
<telepules>Mezokovesd</telepules>
<utca>Hovirag</utca>
<hazszam>11</hazszam>
</cim>
</dolgozo>
<szamla szamlaId="1">
<datum>2020-12-03</datum>
<osszeg>13000000</osszeg>
</szamla>
</autocentrum>
```

Az XMLSchema elkészítése az XML dokumentum alapján:

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

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="qualified">
  <xs:element name="autocentrum">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="auto" type="autoTipus" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="vasarlas" type="vasarlasTipus" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="megrendelo" type="megrendeloTipus" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="rendeles" type="rendelesTipus" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="beszallito" type="beszallitoTipus" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="kereskedes" type="kereskedesTipus" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="dolgozo" type="dolgozoTipus" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="szamla" type="szamlaTipus" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:complexType name="autoTipus">
    <xs:sequence>
      <xs:element name="ar" type="xs:double"/>
      <xs:element name="szin" type="xs:string"/>
      <xs:element name="hajtas" type="xs:string" maxOccurs="unbounded"/>
      <xs:element name="extrak" type="xs:string" maxOccurs="unbounded"/>
      <xs:element name="motor" type="motorTipus"/>
    </xs:sequence>
    <xs:attribute name="autoId" type="xs:integer"/>
  </xs:complexType>

  <xs:complexType name="motorTipus">
    <xs:sequence>
      <xs:element name="hengerurtartalom" type="xs:integer"/>
      <xs:element name="teljesitmeny" type="xs:integer"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="vasarlasTipus">
    <xs:sequence>
      <xs:element name="darabszam" type="xs:integer"/>
      <xs:element name="fizetesimod" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="megrendeloTipus">
    <xs:sequence>
```

```

<xs:element name="nev" type="xs:string"/>
<xs:element name="cim" type="cimTipus"/>
</xs:sequence>
<xs:attribute name="megrendeloId" type="xs:integer"/>
</xs:complexType>

<xs:complexType name="cimTipus">
<xs:sequence>
<xs:element name="iranyitoszam" type="xs:integer"/>
<xs:element name="telepules" type="xs:string"/>
<xs:element name="utca" type="xs:string"/>
<xs:element name="hazszam" type="xs:string"/>
</xs:sequence>
</xs:complexType>

<xs:complexType name="rendelesTipus">
<xs:sequence>
<xs:element name="datum" type="xs:date"/>
<xs:element name="darabszam" type="xs:integer"/>
</xs:sequence>
<xs:attribute name="rendelesId" type="xs:integer"/>
</xs:complexType>

<xs:complexType name="beszallitoTipus">
<xs:sequence>
<xs:element name="elerhetoseg" type="xs:string" maxOccurs="unbounded"/>
<xs:element name="markanev" type="xs:string"/>
<xs:element name="cim" type="cimTipus"/>
</xs:sequence>
<xs:attribute name="beszallitoId" type="xs:integer"/>
</xs:complexType>

<xs:complexType name="kereskedesTipus">
<xs:sequence>
<xs:element name="autojellemzok" type="xs:string"/>
<xs:element name="tartalmaz" type="xs:boolean"/>
</xs:sequence>
<xs:attribute name="kereskedesId" type="xs:integer"/>
</xs:complexType>

<xs:complexType name="dolgozoTipus">
<xs:sequence>
<xs:element name="beosztas" type="xs:string" />
<xs:element name="fizetes" type="xs:integer"/>
<xs:element name="cim" type="cimTipus"/>
</xs:sequence>
<xs:attribute name="dolgozoid" type="xs:integer"/>
</xs:complexType>

<xs:complexType name="szamlaTipus">
<xs:sequence>
<xs:element name="datum" type="xs:date"/>
<xs:element name="osszeg" type="xs:integer"/>
</xs:sequence>
<xs:attribute name="szamlaId" type="xs:integer"/>

```

```
</xs:complexType>
</xs:schema>
```

Az XMLSchema elkészítése során saját típusokat használtam és természetesen minden a két dokumentumot validáltattam IntelliJ fejlesztő környezettel.

2.Feladat

### **Adatolvasás:**

Minden adatot kiolvas az XML dokumentumból és a rendezve megjeleníti a consolon

```
package hu.domparse.bbkhxif;

import java.io.File;
import java.io.IOException;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;

import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;

public class DOMReadBBKHXF {

    public static void main(String[] args) throws
ParserConfigurationException, SAXException, IOException {
        //file nev megadasa
        File file = new File("XMLbbkhxf.xml");

        //kesobb hasznalt valtozok deklaralasa
        String prev="";
        int count=0;
        //parszolas
        DocumentBuilderFactory factory =
DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = factory.newDocumentBuilder();

        Document doc = dBuilder.parse(file);

        doc.getDocumentElement().normalize();
        //gyokerelem kiiratasa
        System.out.println("Root element: " +
doc.getDocumentElement().getnodeName());
        //gyerekelemek mentese listaba
        NodeList nList = (NodeList) doc.getDocumentElement();

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

```

        Node node = nList.item(i);

        if(!node.getNodeName().equals("#text")) {
            System.out.println("\n");
            System.out.println("Current element: " +
node.getNodeName());
        }

        if(node.getNodeType()==Node.ELEMENT_NODE) {
            Element elem = (Element) node;
            //id kiiratasa
            String id =
elem.getAttribute(node.getNodeName()+"Id");
            NodeList childs = elem.getChildNodes();

            if(id!="")
                System.out.println("ID: "+id);
            for (int j = 0; j < childs.getLength() ; j++)
{
                Node node2 = childs.item(j);
                if(!node2.getNodeName().equals("#text"))
{
                    int k = 0;
                    String result = node2.getNodeName() +
:
"+elem.getElementsByTagName(node2.getNodeName()).item(k).getTextCo
ntent();
                    if(prev.equals(result)) {
                        k++;
                        result = node2.getNodeName() + :
"+elem.getElementsByTagName(node2.getNodeName()).item(k).getTextCo
ntent();
                    }
                    if(result.contains("motor")) {
                        Node node3 =
elem.getElementsByTagName("hengerurtartalom").item(0);
                        Node node4 =
elem.getElementsByTagName("teljesitmeny").item(0);
                        result = "motor:
\n"+          "+node3.getNodeName() + : "+node3.getTextContent()
+", \n"+          "+node4.getNodeName() + : "+node4.getTextContent();
                    }
                    if(result.contains("cim")) {
                        Node node3 =
elem.getElementsByTagName("iranyitoszam").item(0);
                        Node node4 =
elem.getElementsByTagName("telepules").item(0);
                        Node node5 =
elem.getElementsByTagName("utca").item(0);
                        Node node6 =
elem.getElementsByTagName("hazszam").item(0);

```

### **Adatmódosítás:**

Az első auto ID-jét 6-ra módosítja, valamint az első 2 autó színeit rendre Sárgára és Pirosra változtatja meg, és kitörli a számlákat, ezután pedig megjeleníti az eredményt a consolon

```
package hu.domparse.bbkhxsf;

import java.io.File;
import java.io.IOException;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;

import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.NamedNodeMap;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;

public class DOMModifyBBKHXF {

    public static void main(String[] args) throws
ParserConfigurationException, SAXException, IOException
    {
        try {
            Document doc = DocumentBuilderFactory.newInstance()
                .newDocumentBuilder().parse(new File("bbkhxsf.xml"));
            Transformer transformer = TransformerFactory.newInstance()
                .newTransformer();
            transformer.transform(new DOMSource(doc),
                new StreamResult(new File("bbkhxsf-modified.xml")));
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```

        //file nev megadasa, parszolas
        File inputFile = new File("XML2bbkhxf.xml");
        DocumentBuilderFactory docFactory =
DocumentBuilderFactory.newInstance();
        DocumentBuilder docBuilder =
docFactory.newDocumentBuilder();
        Document doc = docBuilder.parse(inputFile);

        //az else es a masodik auto adatainak mentese
        Node car =
doc.getElementsByTagName("auto").item(0);
        Node car1 =
doc.getElementsByTagName("auto").item(1);
        //gyokerelem mentese
        Node autocentrum = doc.getFirstChild();

        //autoId modositasa
        NamedNodeMap attr = car.getAttributes();
        Node nodeAttr = attr.getNamedItem("autoId");
        nodeAttr.setTextContent("6");

        //az else auto szinenek modositasa feketerol
sargara
        NodeList list = car.getChildNodes();
        for (int i = 0; i < list.getLength(); i++) {
            Node node = list.item(i);
            if (node.getNodeType() == Node.ELEMENT_NODE) {
                Element elem = (Element) node;
                if ("szin".equals(elem.getnodeName())) {

if("Fekete".equals(elem.getTextContent())) {
                    elem.setTextContent("Sarga");
                }
            }
        }

        //a masodik auto szinenek modositasa feherrol
pirosra
        NodeList list1 = car1.getChildNodes();
        for (int i = 0; i < list1.getLength(); i++) {
            Node node1 = list1.item(i);
            if (node1.getNodeType() == Node.ELEMENT_NODE)
{
                Element elem1 = (Element) node1;
                if ("szin".equals(elem1.getnodeName())) {

if("Feher".equals(elem1.getTextContent())) {
                    elem1.setTextContent("Piros");
                }
            }
        }
    }
}

```

```
        }

        //szamlak kitorlese
        NodeList childNodes = autocentrum.getChildNodes();
        for(int i = 0; i < childNodes.getLength(); i++) {
            Node node = childNodes.item(i);

            if("szamla".equals(node.getNodeName()))
                autocentrum.removeChild(node);
        }

        //megjelenites a consolon
        TransformerFactory transformerFactory =
TransformerFactory.newInstance();
        Transformer transformer =
transformerFactory.newTransformer();
        DOMSource source = new DOMSource(doc);
        System.out.println("-----New
File-----");
        StreamResult consoleResult = new
StreamResult(System.out);
        transformer.transform(source, consoleResult);
    }catch(Exception e) {
        e.printStackTrace();
    }
}
```

## **Adatlekérdezés**

Az XML-ben található auto nevű gyerekelemeket kérdezi le a fájlból:

```
package hu.domparse.bbkhxfs;

import java.io.File;
import java.io.IOException;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;

import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;

public class DOMQueryBBKHXF {
```

```

    public static void main(String[] args) throws
ParserConfigurationException, SAXException, IOException {
    //file nev megadasa
    File file = new File("XML1bbkhxf.xml");
    //parszolas
    DocumentBuilderFactory dbFactory =
DocumentBuilderFactory.newInstance();
    DocumentBuilder dBuilder =
dbFactory.newDocumentBuilder();

    Document doc = dBuilder.parse(file);
    doc.getDocumentElement().normalize();
    //gyokerelem kiiratasa
    System.out.print("Root element: ");

System.out.println(doc.getDocumentElement().getNodeName());
    //autok mentese listaba
    NodeList nList = doc.getElementsByTagName("auto");

    System.out.println("-----");

    //autok es gyerek elemeik megjelenitese a consolon
    for (int i = 0; i < nList.getLength(); i++) {
        Node node = nList.item(i);
        System.out.println("\nCurrent Element :
"+node.getNodeName());
        if (node.getNodeType() == Node.ELEMENT_NODE) {
            Element elem = (Element) node;

System.out.println("ID:"+elem.getAttribute("autoId"));
            NodeList nList2 = elem.getChildNodes();
            for (int j = 0; j < nList2.getLength(); j++) {
                Node node2 = nList2.item(j);
                if (node2.getNodeType() ==
Node.ELEMENT_NODE) {
                    Element elem2 = (Element) node2;
                    if(!
node2.getNodeName().equals("motor")) {

System.out.println(node2.getNodeName()+" :
"+node2.getTextContent());
                }
                NodeList nList3 =
elem2.getChildNodes();
                    for (int k = 0; k <
nList3.getLength(); k++) {
                        Node node3 = nList3.item(k);

if(node3.getNodeType()==Node.ELEMENT_NODE) {
                        Element elem3 = (Element)
node3;

```

```
System.out.println("motor : "+node3.getNodeName()+
"+node3.getTextContent());
}
}
}
}
}
}
```