**JEGYZŐKÖNYV**

Adatkezelés XML környezetben

Féléves feladat

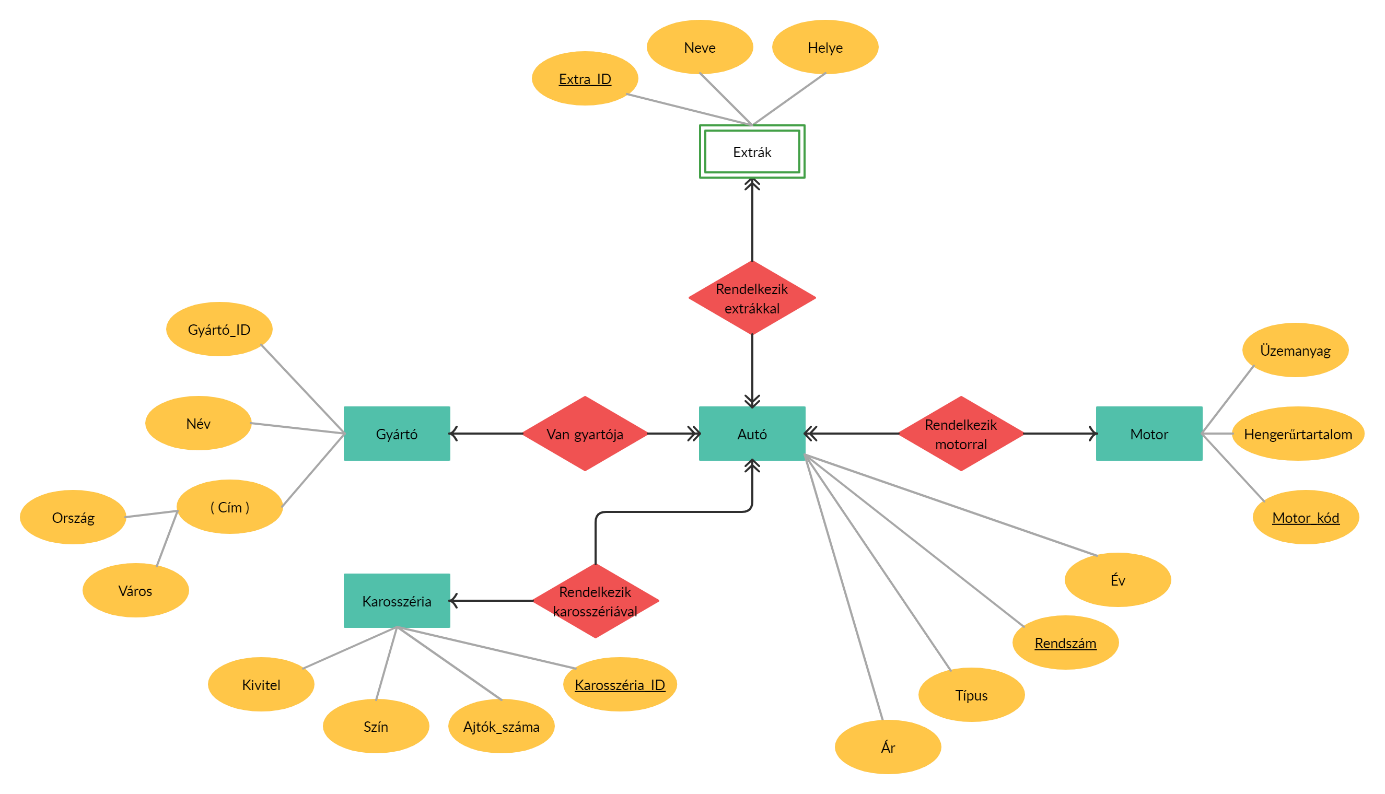
Készítette: Hornyák Balázs Boldizsár

Neptunkód: AD4AYB

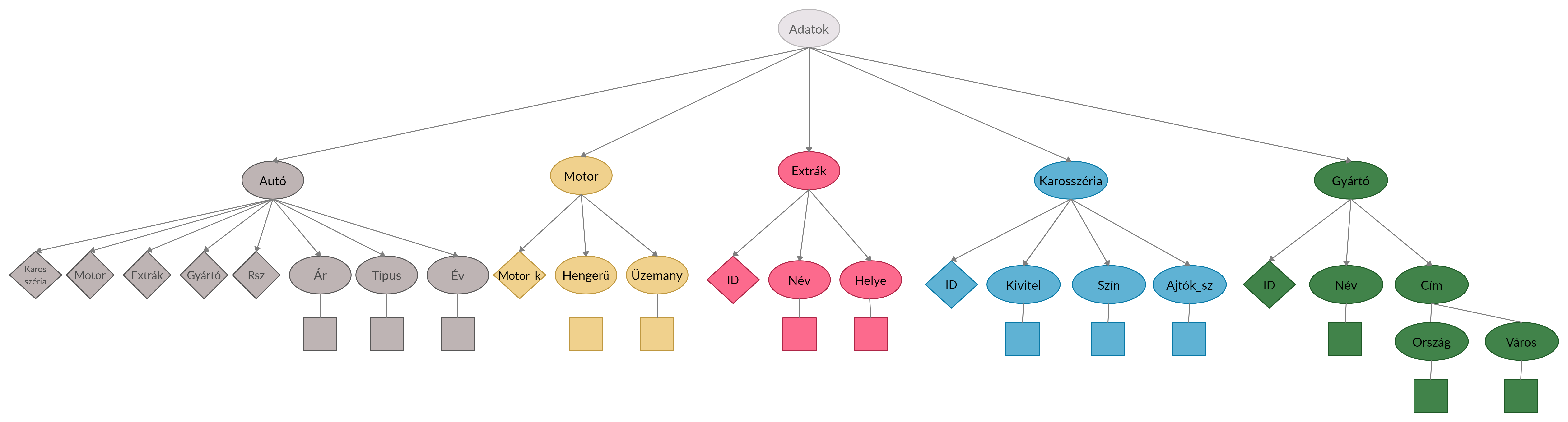
A feladat leírása:

Lorem ipsum is a pseudo-Latin text used in web design, typography, layout, and printing in place of English to emphasise design elements over content. It's also called placeholder (or filler) text. It's a convenient tool for mock-ups. It helps to outline the visual elements of a document or presentation, eg typography, font, or layout. Lorem ipsum is mostly a part of a Latin text by the classical author and philosopher Cicero. Its words and letters have been changed by addition or removal, so to deliberately render its content nonsensical; it's not genuine, correct, or comprehensible Latin anymore. While lorem ipsum's still resembles classical Latin, it actually has no meaning whatsoever. As Cicero's text doesn't contain the letters K, W, or Z, alien to latin, these, and others are often inserted randomly to mimic the typographic appearence of European languages, as are digraphs not to be found in the original. In a professional context it often happens that private or corporate clients corder a publication to be made and presented with the actual content still not being ready. Think of a news blog that's filled with content hourly on the day of going live. However, reviewers tend to be distracted by comprehensible content, say, a random text copied from a newspaper or the internet. The are likely to focus on the text, disregarding the layout and its elements. Besides, random text risks to be unintendedly humorous or offensive, an unacceptable risk in corporate environments. Lorem ipsum and its many variants have been employed since the early 1960ies, and quite likely since the sixteenth century.

**1 feladat**

**1a) Az adatbázis ER modell: **

**1b) Az adatbázis konvertálása XDM modellre:**



**1c) Az XDM modell alapján XML dokumentum készítése:**

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

<nyilvantartas xmlns="nyilvantartas" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="nyilvantartas XMLSchemaAd4ayb.xsd">

<auto rsz="ABC123" motorid="a01" extraid="001 002 004" karosszeriaid="au1" gyartoid="101">

<ev>2016</ev>

<tipus>A6</tipus>

<ár>12000000</ár>

</auto>

<auto rsz="ABC345" motorid="a02" extraid="001 002 003 004" karosszeriaid="au2" gyartoid="101">

<ev>2018</ev>

<tipus>A8</tipus>

<ár>21000000</ár>

</auto>

<auto rsz="XYZ345" motorid="a01" extraid="001" karosszeriaid="au3" gyartoid="101">

<ev>2010</ev>

<tipus>Q3</tipus>

<ár>6000000</ár>

</auto>

<auto rsz="KLE345" motorid="o01" karosszeriaid="op1" gyartoid="102">

<ev>2017</ev>

<tipus>Astra</tipus>

<ár>4500000</ár>

</auto>

<auto rsz="KLG111" motorid="o01" extraid="001 003" karosszeriaid="op1" gyartoid="102">

<ev>2018</ev>

<tipus>Astra</tipus>

<ár>5500000</ár>

</auto>

<auto rsz="SSS045" motorid="s01" extraid="001" karosszeriaid="sk1" gyartoid="103">

<ev>2019</ev>

<tipus>Octavia</tipus>

<ár>7500000</ár>

</auto>

<auto rsz="SZX085" motorid="s02" extraid="001 002 003" karosszeriaid="sk2" gyartoid="103">

<ev>2020</ev>

<tipus>Superb</tipus>

<ár>11500000</ár>

</auto>

<motor id="a01">

<uzemanyag>diesel</uzemanyag>

<cc>3000</cc>

</motor>

<motor id="a02">

<uzemanyag>benzin</uzemanyag>

<cc>4500</cc>

</motor>

<motor id="o01">

<uzemanyag>benzin</uzemanyag>

<cc>2000</cc>

</motor>

<motor id="s01">

<uzemanyag>benzin</uzemanyag>

<cc>1600</cc>

</motor>

<motor id="s02">

<uzemanyag>diesel</uzemanyag>

<cc>2000</cc>

</motor>

<karosszeria id="au1">

<kivitel>kombi</kivitel>

<szin>fekete</szin>

<ajtok\_szama>5</ajtok\_szama>

</karosszeria>

<karosszeria id="au2">

<kivitel>sedan</kivitel>

<szin>feher</szin>

<ajtok\_szama>4</ajtok\_szama>

</karosszeria>

<karosszeria id="au3">

<kivitel>suv</kivitel>

<szin>fekete</szin>

<ajtok\_szama>5</ajtok\_szama>

</karosszeria>

<karosszeria id="op1">

<kivitel>hatchback</kivitel>

<szin>szurke</szin>

<ajtok\_szama>5</ajtok\_szama>

</karosszeria>

<karosszeria id="sk1">

<kivitel>kombi</kivitel>

<szin>kek</szin>

<ajtok\_szama>5</ajtok\_szama>

</karosszeria>

<karosszeria id="sk2">

<kivitel>sedan</kivitel>

<szin>zold</szin>

<ajtok\_szama>4</ajtok\_szama>

</karosszeria>

<extra id="001">

<nev>ablak sotetites</nev>

<hely>kulso</hely>

</extra>

<extra id="002">

<nev>sávtartó</nev>

<hely>kulso</hely>

</extra>

<extra id="003">

<nev>ules futes</nev>

<hely>belso</hely>

</extra>

<extra id="004">

<nev>bor ules</nev>

<hely>belso</hely>

</extra>

<gyarto id="101">

<nev>Audi</nev>

<cim>

<orszag>Nemetorszag</orszag>

<varos>Ingolstadt</varos>

</cim>

</gyarto>

<gyarto id="102">

<nev>Opel</nev>

<cim>

<orszag>Nemetorszag</orszag>

<varos>Rüsselsheim</varos>

</cim>

</gyarto>

<gyarto id="103">

<nev>Skoda</nev>

<cim>

<orszag>Csehország</orszag>

<varos>Mladá Boleslav</varos>

</cim>

</gyarto>

</nyilvantartas>

**1d) Az XML dokumentum alapján XMLSchema készítése:**

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

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" targetNamespace="nyilvantartas" xmlns:ns1="nyilvantartas">

<xs:element name="nyilvantartas">

<xs:complexType mixed="true">

<xs:choice minOccurs="0" maxOccurs="unbounded">

<xs:element ref="ns1:auto"/>

<xs:element ref="ns1:extra"/>

<xs:element ref="ns1:gyarto"/>

<xs:element ref="ns1:karosszeria"/>

<xs:element ref="ns1:motor"/>

</xs:choice>

</xs:complexType>

</xs:element>

<xs:element name="auto">

<xs:complexType>

<xs:sequence>

<xs:element ref="ns1:ev"/>

<xs:element ref="ns1:tipus"/>

<xs:element ref="ns1:ár"/>

</xs:sequence>

<xs:attribute name="extraid"/>

<xs:attribute name="gyartoid" use="required" type="xs:integer"/>

<xs:attribute name="karosszeriaid" use="required" type="xs:NCName"/>

<xs:attribute name="motorid" use="required" type="xs:NCName"/>

<xs:attribute name="rsz" use="required" type="xs:NCName"/>

</xs:complexType>

</xs:element>

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

<xs:element name="tipus" type="xs:NCName"/>

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

<xs:element name="extra">

<xs:complexType>

<xs:sequence>

<xs:element ref="ns1:nev"/>

<xs:element ref="ns1:hely"/>

</xs:sequence>

<xs:attribute name="id" use="required" type="xs:integer"/>

</xs:complexType>

</xs:element>

<xs:element name="hely" type="xs:NCName"/>

<xs:element name="gyarto">

<xs:complexType>

<xs:sequence>

<xs:element ref="ns1:nev"/>

<xs:element ref="ns1:cim"/>

</xs:sequence>

<xs:attribute name="id" use="required" type="xs:integer"/>

</xs:complexType>

</xs:element>

<xs:element name="cim">

<xs:complexType>

<xs:sequence>

<xs:element ref="ns1:orszag"/>

<xs:element ref="ns1:varos"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="orszag" type="xs:NCName"/>

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

<xs:element name="karosszeria">

<xs:complexType>

<xs:sequence>

<xs:element ref="ns1:kivitel"/>

<xs:element ref="ns1:szin"/>

<xs:element ref="ns1:ajtok\_szama"/>

</xs:sequence>

<xs:attribute name="id" use="required" type="xs:NCName"/>

</xs:complexType>

</xs:element>

<xs:element name="kivitel" type="xs:NCName"/>

<xs:element name="szin" type="xs:NCName"/>

<xs:element name="ajtok\_szama" type="xs:integer"/>

<xs:element name="motor">

<xs:complexType>

<xs:sequence>

<xs:element ref="ns1:uzemanyag"/>

<xs:element ref="ns1:cc"/>

</xs:sequence>

<xs:attribute name="id" use="required" type="xs:NCName"/>

</xs:complexType>

</xs:element>

<xs:element name="uzemanyag" type="xs:NCName"/>

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

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

</xs:schema>

**2 feladat**

**2a) adatolvasás**

package hu.domparse.ad4ayb;

import java.io.File;

import java.util.Scanner;

import javax.xml.parsers.DocumentBuilderFactory;

import javax.xml.parsers.DocumentBuilder;

import org.w3c.dom.Document;

import org.w3c.dom.NodeList;

import org.w3c.dom.Node;

import org.w3c.dom.Element;

public class DOMReadAd4ayb {

public static void main(String[] args) {

try {

File inputFile = new File("XMLAd4ayb.xml");

DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();

DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();

Document doc = dBuilder.parse(inputFile);

doc.getDocumentElement().normalize();

//lent csináltam egy "menut" ahol egy switch el lehet a "lépkedni"

menu(doc);

} catch (Exception e) {

e.printStackTrace();

}

}

//kilistazza az osszes autot ami szerepel az XML ben

public static void ListAuto(Document doc) {

System.out.println("\nMinden autó:");

NodeList nList = doc.getElementsByTagName("auto");

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

Node nNode = nList.item(temp);

System.out.println();

if (nNode.getNodeType() == Node.ELEMENT\_NODE) {

Element eElement = (Element) nNode;

System.out.println("rendszam: " +

eElement.getAttribute("rsz"));

System.out.println("ev: " +

eElement

.getElementsByTagName("ev")

.item(0)

.getTextContent());

System.out.println("tipus: " +

eElement

.getElementsByTagName("tipus")

.item(0)

.getTextContent());

System.out.println("ár : " +

eElement

.getElementsByTagName("ar")

.item(0)

.getTextContent() +

"Ft");

}

}

}

//kilistazza az osszes gyartot ami szerepel az XML ben

public static void ListGyarto(Document doc) {

System.out.println("\nMinden gyártó:");

NodeList nList = doc.getElementsByTagName("gyarto");

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

Node nNode = nList.item(temp);

System.out.println();

if (nNode.getNodeType() == Node.ELEMENT\_NODE) {

Element eElement = (Element) nNode;

System.out.println("id: " +

eElement.getAttribute("id"));

System.out.println("nev: " +

eElement

.getElementsByTagName("nev")

.item(0)

.getTextContent());

System.out.println("orszag: " +

eElement

.getElementsByTagName("orszag")

.item(0)

.getTextContent());

System.out.println("varos: " +

eElement

.getElementsByTagName("varos")

.item(0)

.getTextContent());

}

}

}

//kilistazza az osszes karosszeriat ami szerepel az XML ben

public static void ListKarosszeria(Document doc) {

System.out.println("\nMinden karosszeria:");

NodeList nList = doc.getElementsByTagName("karosszeria");

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

Node nNode = nList.item(temp);

System.out.println();

if (nNode.getNodeType() == Node.ELEMENT\_NODE) {

Element eElement = (Element) nNode;

System.out.println("id: " +

eElement.getAttribute("id"));

System.out.println("kivitel: " +

eElement

.getElementsByTagName("kivitel")

.item(0)

.getTextContent());

System.out.println("szin: " +

eElement

.getElementsByTagName("szin")

.item(0)

.getTextContent());

System.out.println("ajtok száma: " +

eElement

.getElementsByTagName("ajtok\_szama")

.item(0)

.getTextContent());

}

}

}

//kilistazza az osszes extrat ami szerepel az XML ben

public static void ListExtrak(Document doc) {

System.out.println("\nMinden extra:");

NodeList nList = doc.getElementsByTagName("extra");

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

Node nNode = nList.item(temp);

System.out.println();

if (nNode.getNodeType() == Node.ELEMENT\_NODE) {

Element eElement = (Element) nNode;

System.out.println("id: " +

eElement.getAttribute("id"));

System.out.println("nev: " +

eElement

.getElementsByTagName("nev")

.item(0)

.getTextContent());

System.out.println("hely: " +

eElement

.getElementsByTagName("hely")

.item(0)

.getTextContent());

}

}

}

//kilistazza az osszes mototrt ami szerepel az XML ben

public static void ListMotor(Document doc) {

System.out.println("\nMinden motor:");

NodeList nList = doc.getElementsByTagName("motor");

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

Node nNode = nList.item(temp);

System.out.println();

if (nNode.getNodeType() == Node.ELEMENT\_NODE) {

Element eElement = (Element) nNode;

System.out.println("id: " +

eElement.getAttribute("id"));

System.out.println("uzemanyag: " +

eElement

.getElementsByTagName("uzemanyag")

.item(0)

.getTextContent());

System.out.println("hengeruertartalom: " +

eElement

.getElementsByTagName("cc")

.item(0)

.getTextContent() +

"cm3");

}

}

}

//ez a main ben hivodik meg, ahol atadom neki a Document-et hogy a switchen belul at tudjam adni az

//adott fugvenynek.

private static void menu(Document doc) {

@SuppressWarnings("resource")

Scanner scann = new Scanner(System.in);

menuText();

//string ként olvasom be így ha más stringet írnak be mint ami a menuben választható,

//akkor a default lép érványbe és kiirja, hogy wrong input, majd megivja megint a menut.

//Minden menu hivas utan megjelenik menint a menu hogy ujabb lekerdeset csinaljunk,

//ha q-t megnyomjunk ki tudunk lepni a programbol

String menu = scann.next();

switch (menu) {

case "1":

{

ListAuto(doc);

System.out.println();

menu(doc);

break;

}

case "2":

{

ListGyarto(doc);

System.out.println();

menu(doc);

break;

}

case "3":

{

ListKarosszeria(doc);

System.out.println();

menu(doc);

break;

}

case "4":

{

ListExtrak(doc);

System.out.println();

menu(doc);

break;

}

case "5":

{

ListMotor(doc);

System.out.println();

menu(doc);

break;

}

case "q":

{

break;

}

default:

System.out.println("Wrong input!");

menu(doc);

break;

}

}

//ahhoz hogy ne kelljen megint leirni a modifyban.

public static void menuText() {

System.out.println("az osszes auto listazasahoz nyomd meg az 1-est!");

System.out.println("az osszes gyarto listazasahoz nyomd meg az 2-est!");

System.out.println("az osszes karosszéria listazasahoz nyomd meg az 3-ast!");

System.out.println("az osszes extrak listazasahoz nyomd meg az 4-est!");

System.out.println("az osszes motor listazasahoz nyomd meg az 5-ost!");

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

System.out.println("kilepeshez nyomja meg a q-t");

}

}

**2 feladat**

**2b) adatmódosítás**

package hu.domparse.ad4ayb;

import java.io.File;

import java.util.Scanner;

import javax.xml.parsers.DocumentBuilder;

import javax.xml.parsers.DocumentBuilderFactory;

import javax.xml.transform.Transformer;

import javax.xml.transform.TransformerException;

import javax.xml.transform.TransformerFactory;

import javax.xml.transform.dom.DOMSource;

import javax.xml.transform.stream.StreamResult;

import org.w3c.dom.Document;

import org.w3c.dom.Element;

import org.w3c.dom.Node;

import org.w3c.dom.NodeList;

public class DOMModifyAd4ayb2 {

public static void main(String[] args) {

try {

File inputFile = new File("XMLAd4ayb.xml");

DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();

DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();

Document doc = dBuilder.parse(inputFile);

doc.getDocumentElement().normalize();

// irtam egy menut a modify hoz valo navigalashoz

menu(doc);

} catch (Exception e) {

e.printStackTrace();

}

}

private static void menu(Document doc) throws TransformerException {

@SuppressWarnings("resource")

Scanner scann = new Scanner(System.in);

//a readben leirt menupontok

System.out.println("Navigaljon el oda ahhol módosítást szeretne véghezvinni, majd írja be annak az id-ját!\n");

DOMReadAd4ayb.menuText();

//string ként olvasom be így ha más stringet írnak be mint ami a menuben választható,

//akkor a default lép érványbe és kiirja, hogy wrong input, majd megivja megint a menut.

//Minden menu hivas utan megjelenik menint a menu hogy ujabb lekerdeset csinaljunk,

//ha q-t megnyomjunk ki tudunk lepni a programbol

String menu = scann.next();

switch (menu) {

case "1":

{

//a readben levo kiiratast hasznalom

DOMReadAd4ayb.ListAuto(doc);

//bekerem a valtoztatashoz szukseges adatokat

String id = valasztas();

String element = valasztas2();

String newElement = valasztas3();

//atadom a bekert adatokat

ModifyAuto(id, element, newElement, doc);

//ismet meghivodik a menu

DOMReadAd4ayb.ListAuto(doc);

menu(doc);

break;

}

case "2":

{

//a readben levo kiiratast hasznalom

DOMReadAd4ayb.ListGyarto(doc);

//bekerem a valtoztatashoz szukseges adatokat

String id = valasztas();

String element = valasztas2();

String newElement = valasztas3();

//atadom a bekert adatokat

ModifyGyarto(id, element, newElement, doc);

System.out.println("megtortent a modositas");

//ismet meghivodik a menu

menu(doc);

break;

}

case "3":

{

//a readben levo kiiratast hasznalom

DOMReadAd4ayb.ListKarosszeria(doc);

//bekerem a valtoztatashoz szukseges adatokat

String id = valasztas();

String element = valasztas2();

String newElement = valasztas3();

//atadom a bekert adatokat

ModifyKarosszeria(id, element, newElement, doc);

System.out.println("megtortent a modositas");

//ismet meghivodik a menu

menu(doc);

break;

}

case "4":

{

//a readben levo kiiratast hasznalom

DOMReadAd4ayb.ListExtrak(doc);

//bekerem a valtoztatashoz szukseges adatokat

String id = valasztas();

String element = valasztas2();

String newElement = valasztas3();

//atadom a bekert adatokat

ModifyExtrak(id, element, newElement, doc);

System.out.println("megtortent a modositas");

//ismet meghivodik a menu

menu(doc);

break;

}

case "5":

{

//a readben levo kiiratast hasznalom

DOMReadAd4ayb.ListMotor(doc);

//bekerem a valtoztatashoz szukseges adatokat

String id = valasztas();

String element = valasztas2();

String newElement = valasztas3();

//atadom a bekert adatokat

ModifyMotor(id, element, newElement, doc);

System.out.println("megtortent a modositas");

//ismet meghivodik a menu

menu(doc);

break;

}

case "q":

{

break;

}

default:

System.out.println("Wrong input!");

menu(doc);

break;

}

}

//motor adat módosítás

private static void ModifyMotor(String id, String element, String newElement, Document doc) throws TransformerException {

NodeList nList = doc.getElementsByTagName("motor");

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

Node nNode = nList.item(temp);

if (nNode.getNodeType() == Node.ELEMENT\_NODE) {

Element eElement = (Element) nNode;

if (eElement.getAttribute("id").equals(id)) {

eElement.getElementsByTagName(element).item(0).setTextContent(newElement);

}

}

}

CreateXML(doc);

}

//extrák adat módosítás

private static void ModifyExtrak(String id, String element, String newElement, Document doc) throws TransformerException {

NodeList nList = doc.getElementsByTagName("extra");

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

Node nNode = nList.item(temp);

if (nNode.getNodeType() == Node.ELEMENT\_NODE) {

Element eElement = (Element) nNode;

if (eElement.getAttribute("id").equals(id)) {

eElement.getElementsByTagName(element).item(0).setTextContent(newElement);

}

}

}

CreateXML(doc);

}

//karosszeria adat módosítás

private static void ModifyKarosszeria(String id, String element, String newElement, Document doc) throws TransformerException {

NodeList nList = doc.getElementsByTagName("karosszeria");

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

Node nNode = nList.item(temp);

if (nNode.getNodeType() == Node.ELEMENT\_NODE) {

Element eElement = (Element) nNode;

if (eElement.getAttribute("id").equals(id)) {

eElement.getElementsByTagName(element).item(0).setTextContent(newElement);

}

}

}

CreateXML(doc);

}

//gyarto adat módosítás

private static void ModifyGyarto(String id, String element, String newElement, Document doc) throws TransformerException {

NodeList nList = doc.getElementsByTagName("gyarto");

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

Node nNode = nList.item(temp);

if (nNode.getNodeType() == Node.ELEMENT\_NODE) {

Element eElement = (Element) nNode;

if (eElement.getAttribute("id").equals(id)) {

eElement.getElementsByTagName(element).item(0).setTextContent(newElement);

}

}

}

CreateXML(doc);

}

//auto adat módosítás

private static void ModifyAuto(String id, String element, String newElement, Document doc) throws TransformerException {

NodeList nList = doc.getElementsByTagName("auto");

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

Node nNode = nList.item(temp);

if (nNode.getNodeType() == Node.ELEMENT\_NODE) {

Element eElement = (Element) nNode;

if (eElement.getAttribute("rsz").equals(id)) {

eElement.getElementsByTagName(element).item(0).setTextContent(newElement);

}

}

}

CreateXML(doc);

}

//uj XML dokumentumba taroljuk a modositasokat(a regiekkel egyutt persze)

public static void CreateXML(Document doc) throws TransformerException {

TransformerFactory transformerFactory = TransformerFactory.newInstance();

Transformer transformer = transformerFactory.newTransformer();

DOMSource source = new DOMSource(doc);

StreamResult result = new StreamResult(new File("XMLAd4ayb.updated.xml"));

transformer.transform(source, result);

}

//bekérés

private static String valasztas() {

System.out.println("irja az id-t ahol modositani szeretne!");

@SuppressWarnings("resource")

Scanner scann = new Scanner(System.in);

String menu = scann.next();

return menu;

}

//bekérés

private static String valasztas2() {

System.out.println("addja meg azt az adattagot, amit modositani szeretne!");

@SuppressWarnings("resource")

Scanner scann = new Scanner(System.in);

String menu = scann.next();

return menu;

}

//bekérés

private static String valasztas3() {

System.out.println("addja meg, hogy mire modositana!");

@SuppressWarnings("resource")

Scanner scann = new Scanner(System.in);

String menu = scann.next();

return menu;

}

}