

# JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Időjárás megfigyelő rendszer felépítése

Készítette: **Kiss Gergő**

Neptunkód: **BKUPJ9**

**A feladat leírása:** Egy időjárás megfigyelő és előrejelző állomás felépítését fogom lemodellezni, mely jelenidő megfigyeléssel és előrejelzés-algoritmizálással foglalkozik.

Maga a rendszer 4 entitásból áll, melyek a következők: szenzorállomás, elosztóközpont, előrejelzőállomás, monitorozás.

A szenzorállomás figyeli az időjárási adatokat szenzorokon keresztül, kapcsolódik hozzá a SzenzorID attribútum, ezzel lehet beazonosítani. A szenzorok a következők:

- szélirány és szélesség, ebből származtatott adat a szél
- légnyomás
- páratartalom
- hőmérséklet

Az elosztóközpont vezérli az adatáramlást. Tartozik hozzá az elosztandó adat, valamint egy központID.

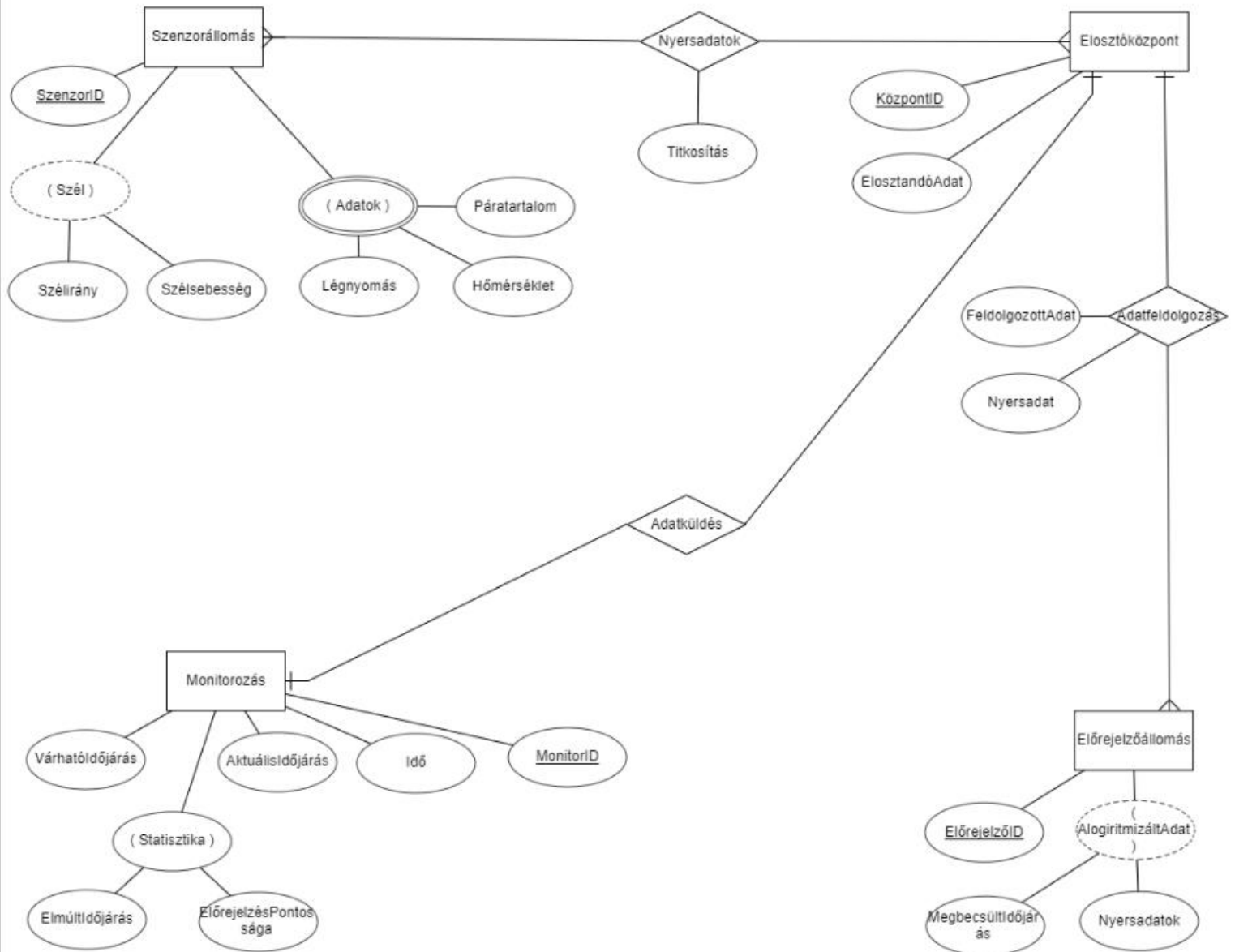
Kapcsolatban áll a 3 másik entitással, ezen kapcsolatok nevei:

- nyersadatok (titkosított adat)
- adatfeldolgozás (feldolgozottAdat és nyersadat)
- adatküldés a monitorozáshoz

A következő az előrejelzőállomás, mely a beérkezett adatok alapján készíti el megadott algoritmusok alapján az előrejelzést. Az elosztóközponttal áll kapcsolatban. Az előrejelzőID alapján azonosítható be.

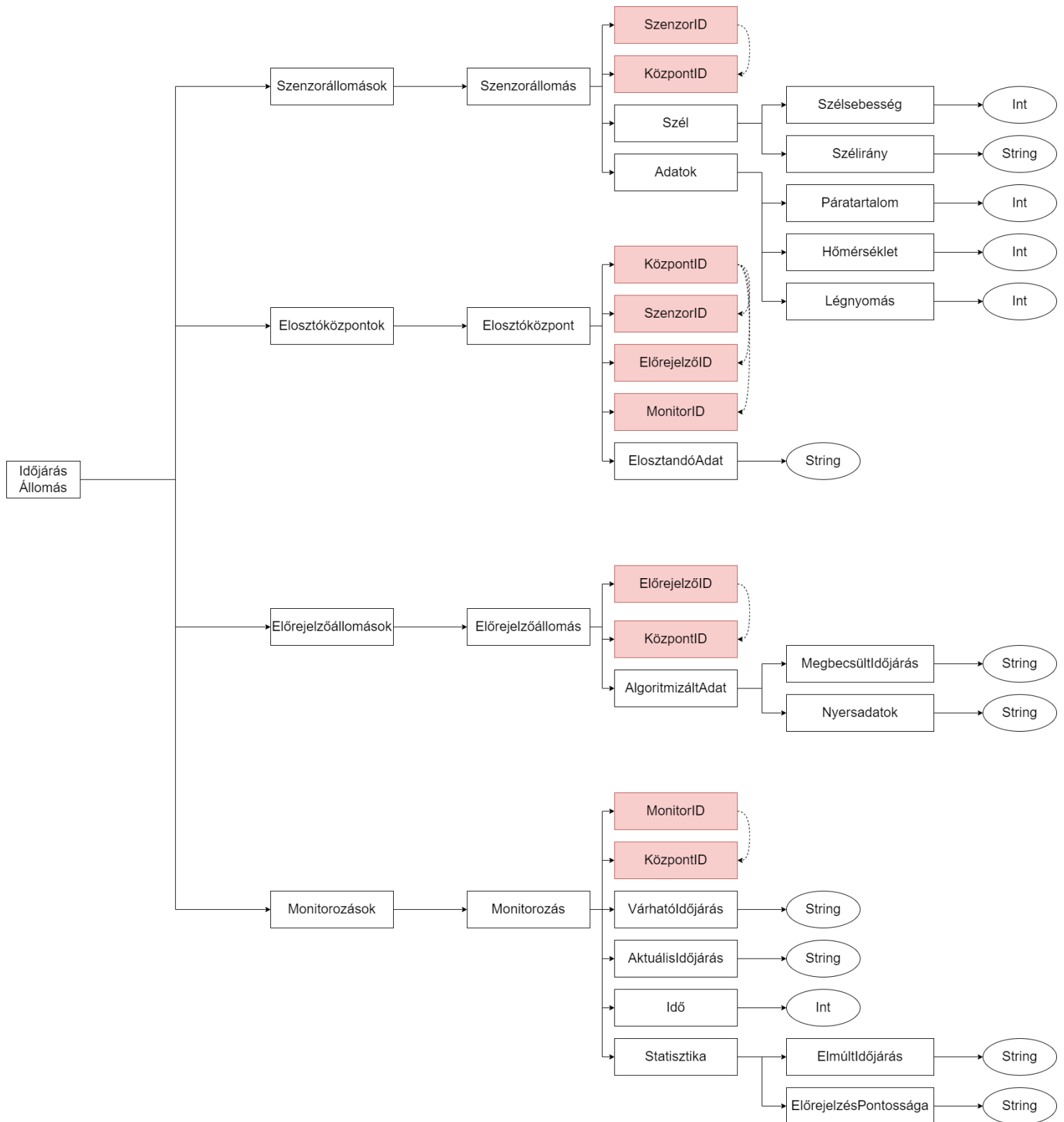
Az utolsó entitás a monitorozóegység, amely különböző adatokat jelez ki. Ezek az adatok: várható időjárás, aktuális időjárás, statisztika (múltbéli időjárás és az előrejelzés pontossága alkotja), valamint az aktuális idő.

## ER modell:



## XDM modell:

Az alábbi XDM modell bemutatja a rendszer felépítését, az egyedeket, az attribútumokat azok típusival együtt.



## XML dokumentum:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <idojarasallomas_adatok xmlns:xsi="http://www.w3.org/2001/XMLSchema" xsi:noNamespaceSchemaLocation="XMLSchemaBKUPJ9.xsd">
3
4 <szenzorallomasok>
5 <szenzorallomas szenzorID="01" kozpontID="01">
6 <szel>
7 <szelirany>keleti</szelirany>
8 <szelsebesség>34</szelsebesség>
9 </szel>
10 <adatok>
11 <páratartalom>87</páratartalom>
12 <hőmérséklet>22</hőmérséklet>
13 <légnyomás>1011</légnyomás>
14 </adatok>
15 </szenzorallomas>
16
17 <szenzorallomas szenzorID="02" kozpontID="02">
18 <szel>
19 <szelirany>nyugati</szelirany>
20 <szelsebesség>12</szelsebesség>
21 </szel>
22 <adatok>
23 <páratartalom>67</páratartalom>
24 <hőmérséklet>18</hőmérséklet>
25 <légnyomás>1001</légnyomás>
26 </adatok>
27 </szenzorallomas>
28 </szenzorallomasok>
```

```
30 <elosztóközpontok>
31 <elosztóközpont kozpontID="01" szenzorID="01" előrejelzőID="01" monitorID="01">
32 <elosztandóAdat>szenzoradatok</elosztandóAdat>
33 </elosztóközpont>
34
35 <elosztóközpont kozpontID="02" szenzorID="02" előrejelzőID="02" monitorID="02">
36 <elosztandóAdat>algoritmizáltadat</elosztandóAdat>
37 </elosztóközpont>
38 </elosztóközpontok>
39
40 <előrejelzőállomások>
41 <előrejelzőállomás előrejelzőID="01" kozpontID="01">
42 <algoritmizáltAdat>
43 <megbecsültIdőjárás>Napos, enyhén szeles, kellemesen meleg</megbecsültIdőjárás>
44 <nyersadatok>Keleti, 34, 87, 22, 1011</nyersadatok>
45 </algoritmizáltAdat>
46 </előrejelzőállomás>
47
48 <előrejelzőállomás előrejelzőID="02" kozpontID="01">
49 <algoritmizáltAdat>
50 <megbecsültIdőjárás>Szeles, borús, hűvös</megbecsültIdőjárás>
51 <nyersadatok>Nyugati, 12, 67, 18, 1001</nyersadatok>
52 </algoritmizáltAdat>
53 </előrejelzőállomás>
54 </előrejelzőállomások>
```

```

56     <monitorozások>
57         <monitorozás monitorID="01" központId="01">
58             <várhatóIdőjárás>Napos, enyhén szeles, kellemesen meleg</várhatóIdőjárás>
59             <aktuálisIdőjárás>Meleg, száraz</aktuálisIdőjárás>
60             <idő>12:20</idő>
61             <statisztika>
62                 <elmúltIdőjárás>melegebb az átlagosnál</elmúltIdőjárás>
63                 <előrejelzésPontossága>pontos</előrejelzésPontossága>
64             </statisztika>
65         </monitorozás>
66
67         <monitorozás monitorID="02" központId="02">
68             <várhatóIdőjárás>Szeles, borús, hűvös</várhatóIdőjárás>
69             <aktuálisIdőjárás>Hűvös, esős</aktuálisIdőjárás>
70             <idő>14:10</idő>
71             <statisztika>
72                 <elmúltIdőjárás>átlagosnál hűvösebb</elmúltIdőjárás>
73                 <előrejelzésPontossága>pontos</előrejelzésPontossága>
74             </statisztika>
75         </monitorozás>
76     </monitorozások>
77
78 </idojarasallomas_adatok>

```

## XML schema:

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="qualified">
3      <xs:element name="idojarasallomas_adatok" type="idojarasallomasAdatokTípus">
4      </xs:element>
5      <xs:complexType name="idojarasallomasAdatokTípus">
6          <xs:sequence>
7              <xs:element name="szensorállomás" type="szensorállomásTípus" minOccurs="0" maxOccurs="unbounded" />
8              <xs:element name="elosztóközpont" type="elosztóközpontTípus" minOccurs="0" maxOccurs="unbounded" />
9              <xs:element name="előrejelzőállomás" type="előrejelzőállomásTípus" minOccurs="0" maxOccurs="unbounded" />
10             <xs:element name="monitorozás" type="monitorozásTípus" minOccurs="0" maxOccurs="unbounded" />
11          </xs:sequence>
12      </xs:complexType>
13
14      <xs:complexType name="szensorállomásTípus">
15          <xs:sequence>
16              <xs:element name="szensorID" type="xs:int"/>
17              <xs:element name="szél" type="szélTípus"/>
18              <xs:element name="adatok" type="adatokTípus"/>
19          </xs:sequence>
20      </xs:complexType>
21
22      <xs:complexType name="elosztóközpontTípus">
23          <xs:sequence>
24              <xs:element name="központID" type="xs:int"/>
25              <xs:element name="elosztandóAdat" type="xs:string"/>
26          </xs:sequence>
27      </xs:complexType>
28

```

```

29     <xs:complexType name="előrejelzőállomásTípus">
30         <xs:sequence>
31             <xs:element name="előrejelzőID" type="xs:int"/>
32             <xs:element name="algoritmizáltAdat" type="algoritmizáltAdatTípus"/>
33         </xs:sequence>
34     </xs:complexType>
35
36     <xs:complexType name="monitorozásTípus">
37         <xs:sequence>
38             <xs:element name="monitorID" type="xs:int"/>
39             <xs:element name="várhatóIdőjárás" type="xs:string"/>
40             <xs:element name="aktuálisIdőjárás" type="xs:string"/>
41             <xs:element name="idő" type="xs:int"/>
42             <xs:element name="statisztika" type="statisztikaTípus"/>
43         </xs:sequence>
44     </xs:complexType>
45
46     <xs:complexType name="szélTípus">
47         <xs:sequence>
48             <xs:element name="szélirány" type="xs:string"/>
49             <xs:element name="szélsebesség" type="xs:int"/>
50         </xs:sequence>
51     </xs:complexType>

```

```

53     <xs:complexType name="adatokTípus">
54         <xs:sequence>
55             <xs:element name="páratartalom" type="xs:int"/>
56             <xs:element name="hőmérséklet" type="xs:int"/>
57             <xs:element name="légnyomás" type="xs:int"/>
58         </xs:sequence>
59     </xs:complexType>
60
61     <xs:complexType name="algoritmizáltAdatTípus">
62         <xs:sequence>
63             <xs:element name="megbecsültIdőjárás" type="xs:string"/>
64             <xs:element name="nyersadatok" type="xs:string"/>
65         </xs:sequence>
66     </xs:complexType>
67
68     <xs:complexType name="statisztikaTípus">
69         <xs:sequence>
70             <xs:element name="elmúltIdőjárás" type="xs:string"/>
71             <xs:element name="előrejelzésPontossága" type="xs:string"/>
72         </xs:sequence>
73     </xs:complexType>
74 </xs:schema>

```

# Adatolvasás:

```
1 package hu.domparse.bkupj9;
2
3 import org.w3c.dom.Document;
4 import org.w3c.dom.Element;
5 import org.w3c.dom.Node;
6 import org.w3c.dom.NodeList;
7 import org.xml.sax.SAXException;
8
9 import javax.xml.parsers.DocumentBuilder;
10 import javax.xml.parsers.DocumentBuilderFactory;
11 import javax.xml.parsers.ParserConfigurationException;
12
13 import java.io.File;
14 import java.io.IOException;
15
16 public class DOMReadBKUPJ9 {
17
18     public static void main(String[] args)
19
20         throws ParserConfigurationException, IOException, SAXException {
21
22         File xml = new File("XMLBKUPJ9.xml"); //Az XML file beolvasása
23         DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance(); //DocumentBuilderFactory inicializálása
24
25         DocumentBuilder docBuilder = factory.newDocumentBuilder(); // A builder példányosítása
26         Document document = docBuilder.parse(xml); //A dokumentum konvertálása hogy értse az xml-t
27         document.getDocumentElement().normalize();
28
29         ReadFile(document); //Kíratjuk az összes adatot
30     }
31
32     public static void ReadFile(Document document) { //Az adatok beolvasása és kiírása
33
34         NodeList nodeList1 = document.getElementsByTagName("szenzorállomás"); //Lista létrehozása a gyermekelemeknek a szenzorállomás kulcsszóra
35         for (int i = 0; i < nodeList1.getLength(); i++) {
36             Node szNode = nodeList1.item(i);
37
38             if (szNode.getNodeType() == Node.ELEMENT_NODE) { //Elemek id-hez adása
39                 Element element1 = (Element) szNode;
40                 String kozpont_id = element1.getAttribute("központID"); //Attribútumok eltárolása
41                 String szenzor_id = element1.getAttribute("szenzorID");
42
43                 Node szNode1 = element1.getElementsByTagName("szélirány").item(0); //Elemek eltárolása
44                 String szelirany = szNode1.getTextContent();
45
46                 Node szNode2 = element1.getElementsByTagName("szélesség").item(0);
47                 String szelsebesség = szNode2.getTextContent();
48
49                 Node szNode3 = element1.getElementsByTagName("páratartalom").item(0);
50                 String paratartalom = szNode3.getTextContent();
51
52                 Node szNode4 = element1.getElementsByTagName("hőmérséklet").item(0);
53                 String homerseklet = szNode4.getTextContent();
54
55                 Node szNode5 = element1.getElementsByTagName("légnyomás").item(0);
56                 String legnyomas = szNode5.getTextContent();
57
58                 System.out.println("Szenzorállomás adatok: Központ ID: " + kozpont_id + ", Szenzor ID: " + szenzor_id + ", szélirány: "
59                     + szelirany + ", szélesség: " + szelsebesség + ", páratartalom: " + paratartalom + ", hőmérséklet: " + homerseklet + "."); //Az eltárolt adatok kiírása
60             }
61         }
62
63         NodeList nodeList2 = document.getElementsByTagName("elosztóközpont");
64         for (int i = 0; i < nodeList2.getLength(); i++) {
65             Node eloNode = nodeList2.item(i);
66
67             if (eloNode.getNodeType() == Node.ELEMENT_NODE) {
68                 Element element2 = (Element) eloNode;
69                 String elorejelzo_id = element2.getAttribute("előrejelzőID");
70                 String kozpont_id = element2.getAttribute("központID");
71                 String monitor_id = element2.getAttribute("monitorID");
72                 String szenzor_id = element2.getAttribute("szenzorID");
73
74                 Node eloNode1 = element2.getElementsByTagName("elosztandóAdat").item(0);
75                 String elosztandoAdat = eloNode1.getTextContent();
76
77                 System.out.println("Elosztóközpont adatok: előrejelző ID: " + elorejelzo_id + ", Központ ID: " + kozpont_id + ", Monitor ID: "
78                     + monitor_id + ", Szenzor ID: " + szenzor_id + ", ElosztandóAdat: " + elosztandoAdat);
79             }
80         }
81
82         NodeList nodeList3 = document.getElementsByTagName("előrejelzőállomás");
83         for (int i = 0; i < nodeList3.getLength(); i++) {
84             Node elorNode = nodeList3.item(i);
85
86             if (elorNode.getNodeType() == Node.ELEMENT_NODE) {
87                 Element element3 = (Element) elorNode;
88                 String elorejelzo_id = element3.getAttribute("előrejelzőID");
89                 String kozpont_id = element3.getAttribute("központID");
90
91                 Node elorNode1 = element3.getElementsByTagName("megbecsültIdőjárás").item(0);
92                 String megbecsult_idojaras = elorNode1.getTextContent();
93
94                 Node elorNode2 = element3.getElementsByTagName("nyersadatok").item(0);
95                 String nyersadatok = elorNode2.getTextContent();
96             }
97         }
98     }
99 }
```



```

97         System.out.println("Előrejelzőállomás adatok: Előrejelző ID: " + előrejelzo_id + ", TKözpont ID " + kozpont_id + ", Megbecsült Időjárás"
98         + megbecsult_idojaras + ", Nyersadatok: " + nyersadatok);
99     }
100 }
101
102
103 NodeList nodeList4 = document.getElementsByTagName("monitorozás");
104 for (int i = 0; i < nodeList4.getLength(); i++) {
105     Node mNode = nodeList4.item(i);
106
107     if (mNode.getNodeType() == Node.ELEMENT_NODE) {
108         Element element4 = (Element) mNode;
109         String kozpont_id = element4.getAttribute("központId");
110         String monitor_id = element4.getAttribute("monitorID");
111
112         Node mNode1 = element4.getElementsByTagName("várhatóIdőjárás").item(0);
113         String varhatoIdojaras = mNode1.getTextContent();
114
115         Node mNode2 = element4.getElementsByTagName("aktuálisIdőjárás").item(0);
116         String aktualisIdojaras = mNode2.getTextContent();
117
118         Node mNode3 = element4.getElementsByTagName("idő").item(0);
119         String ido = mNode3.getTextContent();
120
121         Node mNode4 = element4.getElementsByTagName("elmúltIdőjárás").item(0);
122         String elmultIdojaras = mNode4.getTextContent();
123
124         Node mNode5 = element4.getElementsByTagName("előrejelzésPontossága").item(0);
125         String előrejelzesPontossaga = mNode5.getTextContent();
126
127         System.out.println("Monitorozás adatok: Központ ID: " + kozpont_id + ", Monitor ID: " + monitor_id +
128         ", Várható Időjárás: " + varhatoIdojaras + ", Aktuális Időjárás: " + aktualisIdojaras + ", GIDő: " + ido +
129         ", ElmultIdőjárás: " + elmultIdojaras + ".");
130     }
131 }
132
133
134 }
135
136 }
137

```

## Adatlekérdezés:

```

1 package hu.domparsa.bkupj9;
2
3 import java.io.File;
4 import java.io.IOException;
5
6 import javax.xml.parsers.DocumentBuilder;
7 import javax.xml.parsers.DocumentBuilderFactory;
8 import javax.xml.parsers.ParserConfigurationException;
9
10 import org.w3c.dom.Document;
11 import org.w3c.dom.Element;
12 import org.w3c.dom.Node;
13 import org.w3c.dom.NodeList;
14 import org.xml.sax.SAXException;
15
16 public class DOMQueryBKUPJ9 {
17
18     public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException {
19
20         File file = new File("XMLBKUPJ9.xml");//Az XML file beolvasása
21
22         DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();//DocumentBuilderFactory inicializálása
23         DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();//A builder példányosítása
24
25         Document doc = dBuilder.parse(file);//A dokumentum konvertálása hogy ertse az xml-t
26         doc.getDocumentElement().normalize();
27         System.out.print("Gyökér element: ");//a gyökerelem kiírása
28         System.out.println(doc.getDocumentElement().getNodeName());
29
30         NodeList nList = doc.getElementsByTagName("szenzorállomás");//szenzorállomások mentése listába
31
32
33         for (int i = 0; i < nList.getLength(); i++) {//szenzorállomás gyerek elemeinek megjelenítése a konzolon
34             Node node = nList.item(i);
35             System.out.println("\nAktuális Elem : "+node.getNodeName());
36             if (node.getNodeType() == Node.ELEMENT_NODE) {
37                 Element elem = (Element) node;
38                 System.out.println("ID: "+elem.getAttribute("szenzorID"));
39                 NodeList nList2 = elem.getChildNodes();
40                 for (int j = 0; j < nList2.getLength(); j++) {
41                     Node node2 = nList2.item(j);
42                     if (node2.getNodeType() == Node.ELEMENT_NODE) {
43                         Element elem2 = (Element) node2;
44                         NodeList nList3 = elem2.getChildNodes();
45                         for (int k = 0; k < nList3.getLength(); k++) {
46                             Node node3 = nList3.item(k);
47                             if (node3.getNodeType() == Node.ELEMENT_NODE) {
48                                 Element elem3 = (Element) node3;
49                                 System.out.println(node3.getNodeName()+" : "+node3.getTextContent());
50                             }
51                         }
52                     }
53                 }
54             }
55         }
56     }
57 }
58

```

## Adatmódosítás:

```
1 package hu.domparse.bkupj9;
2
3 import java.io.*;
4
5 import javax.xml.transform.dom.DOMSource;
6 import javax.xml.transform.stream.StreamResult;
7 import javax.xml.parsers.*;
8 import javax.xml.xpath.*;
9 import javax.xml.transform.Transformer;
10 import javax.xml.transform.TransformerFactory;
11 import javax.xml.transform.TransformerException;
12
13 import org.w3c.dom.*;
14 import org.xml.sax.*;
15
16 public class DOMModifyBKUPJ9 {
17
18     public static void main(String[] args) throws ParserConfigurationException, IOException, SAXException, XPathExpressionException, TransformerException { {
19
20         String xmlFilePath = "XMLBKUPJ9.xml"; //File beolvasása
21         DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
22         DocumentBuilder builder = factory.newDocumentBuilder();
23         Document document = builder.parse(new File(xmlFilePath));
24
25         Node masodikMonitorozs = document.getElementsByTagName("monitorozs").item(1); //A 2. monitorozs elem gyerekelemeinek megkeresése
26         NodeList masodikMonitorozsNodeList = masodikMonitorozs.getChildNodes();
27
28         for (int i = 0; i < masodikMonitorozsNodeList.getLength(); i++) { //A 2. monitorozs elem aktuális id-jének megváltoztatása naposra
29             Node element = masodikMonitorozsNodeList.item(i);
30             if ("aktualisId".equals(element.getNodeName())) {
31                 element.setTextContent("napos");
32             }
33         }
34
35         Element monitorozo = document.createElement("monitoroz"); //A 2. monitorozs elemhez hozzáadjuk a monitoroz nevét
36         monitorozo.appendChild(document.createTextNode("Bla"));
37         masodikMonitorozs.appendChild(monitorozo);
38
39         TransformerFactory transformerFactory = TransformerFactory.newInstance(); //A módosításokat feltöltjük az XML file-ba
40         Transformer transformer = transformerFactory.newTransformer();
41         DOMSource source = new DOMSource(document);
42         StreamResult result = new StreamResult(new File(xmlFilePath));
43         transformer.transform(source, result);
44
45         System.out.println("Változtatások sikeresen eszközölve az XML file-on!!!!"); //Nyugtazzuk az elvégzett módosításokat
46     }
47 }
48 }
49 }
```