JEGYZŐKÖNYV

Webes adatkezelő környezetek

Féléves feladat

Vállalat

Készítette: **Szalóczy Krisztián**

Neptunkód: **Y4O4X0**

Dátum: 2024.12.09

**Miskolc, 2024**

**Tartalomjegyzék**

[Bevezetés: 3](#_Toc183091569)

[Feladat leírása 3](#_Toc183091570)

[1 Adatbázis ER modell készítése 3](#_Toc183091571)

[1.1 Az egyedek tulajdonságai és az ER modell 3](#_Toc183091572)

[1.2 Az adatbázis konvertálása XDM modellre 5](#_Toc183091573)

[1.3 Az XDM modell alapján XML dokumentum készítése 6](#_Toc183091574)

[1.4 Az XML dokumentum alapján XMLSchema készítése 14](#_Toc183091575)

[2. DOM program készítése JAVA környezetbe 20](#_Toc183091576)

[2.1 DOM adatolvasás 20](#_Toc183091577)

[2.2 DOM adatírás 22](#_Toc183091578)

[2.3 DOM adatlekérdezés 24](#_Toc183091579)

[2.4 DOM adatmódosítás 25](#_Toc183091580)

# Bevezetés:

Ez a beadandó a vállalati szervezeti struktúrák **XML** alapú modellezését mutatja be. A rendszer különféle szervezeti elemeket ábrázol, mint a csapatok, dolgozók, részlegek, projektek és ügyfelek, valamint ezek közötti kapcsolatokat. Az **XML** használata biztosítja az adatok strukturált és platformfüggetlen tárolását, ami támogatja a szervezet működésének átlátható kezelését és az adatkapcsolatok egyszerű kereshetőségét. A modell célja, hogy a vállalati hierarchia, valamint az alkalmazotti és projektbeli viszonyok egyértelműen megjelenjenek, így lehetővé téve az adatok könnyű feldolgozását és elemzését.

# Feladat leírása

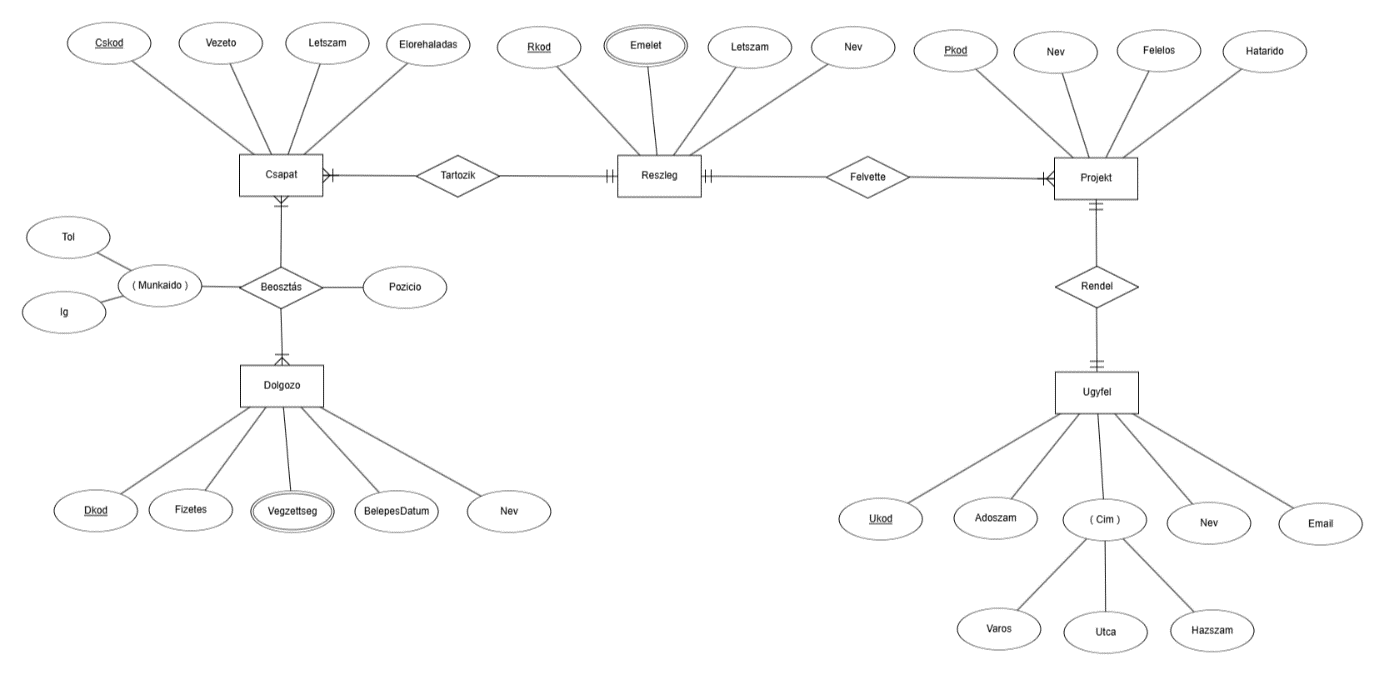
A féléves beadandóm tematikája egy vállalati szervezeti struktúra adatmodelljének létrehozása és **XML** formátumban való megvalósítása. A projekt az egyes szervezeti elemeket – mint például a csapatok, dolgozók, részlegek és projektek – és ezek kapcsolatát modellezi. Az adatmodellben a csapatokhoz csapattagként dolgozók tartoznak, a dolgozók különböző pozíciókat töltenek be, és meghatározott munkaidővel rendelkeznek. A részlegek projektet vehetnek fel, míg az ügyfelek különféle projekteket rendelhetnek. Az **XML** formátum lehetővé teszi az adatok könnyen kereshető és áttekinthető tárolását, amely az adatelemek közötti kapcsolatok egyszerű kezelését biztosítja. Az **XML** dokumentumban tudunk beolvasni, írni, lekérdezni és modosítani.

# 1 Adatbázis ER modell készítése

Az **ER** modell elkészítéséhez az **ERDPlus** online modellkészítő eszközt használtam, amely lehetővé teszi az entitások, attribútumok és kapcsolatok könnyű és gyors ábrázolását. Az **ERDPlus** felülete segítségével hatékonyan létre tudtam hozni a vállalati szervezeti struktúrát bemutató diagramot, amely jól szemlélteti a különböző elemek közötti kapcsolatokat.

## Az egyedek tulajdonságai és az ER modell

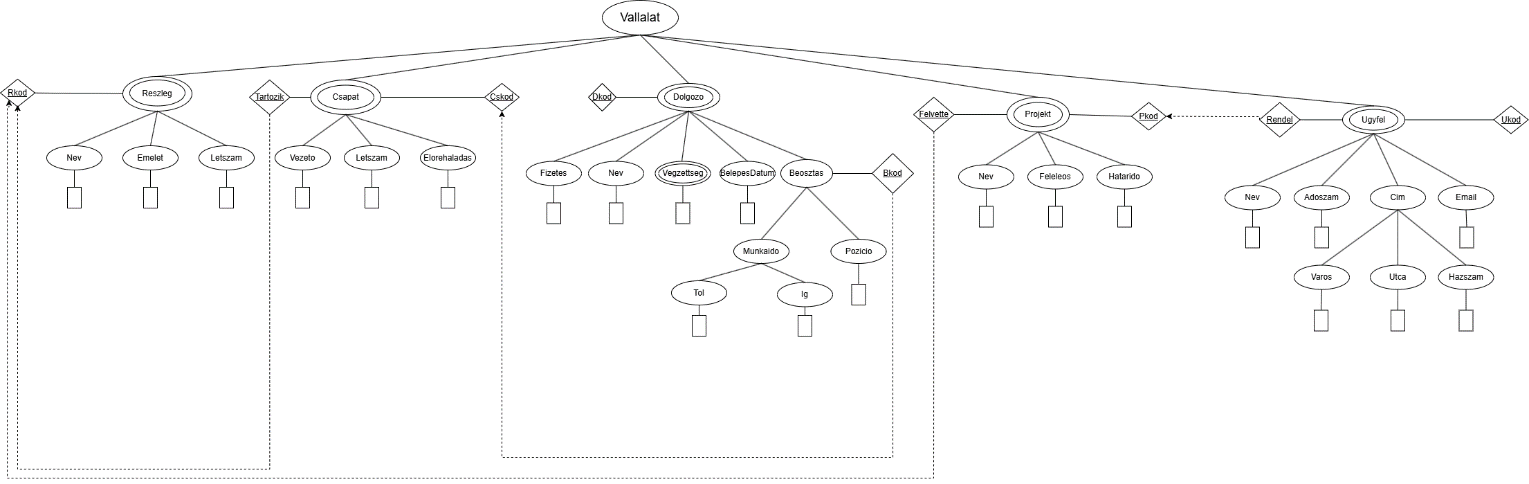
* **Részleg:**
  + **Rkod**: A Részleg egyed elsődleges kulcsa
  + **Emelet:** Emelet amellyen a részleg található
  + **Létszám:** Részleghez tartoró dolgozók létszáma
  + **Név:** Részleg megnevezése
* **Csapat**
  + **Cskod:** A Csapat egyed elsődleges kulcsa
  + **Vezeto:** A csapat vezetője
  + **Létszam:** A csapat létszáma
  + **Előrehaladás :** A projekt teljesítettsége, százalékos érték
* **Beosztás:**
  + **Bkod:** A beosztás egyed elsődleges kulcsa
  + **Munkaidő:** Azadott dolgozó munkarendje**,** összetett elem
    - **Tol:** Munkakezdése
    - **Ig:** Munka befejezése
  + **Pozíció:** A dolgozó szerepe a csapatban
* **Dolgozo:**
  + **Dkod:** Dolgozó egyed elsődleges kulcsa
  + **Fizetés:** A dolgozo fizetése
  + **Végzettség:** A dolgozo végzettsége
  + **BelépésDátum:** A dolgozo belépsi dátumba a vállalatba
* **Projekt:**
  + **Pkod:** Projekt egyed elsődleges kulcsa
  + **Név:** A projekt elnevezése
  + **Felelős**: A személy akihez tartozik a projekt
  + **Határidő**: A projekt befejezésének határideje
* **Ügyfél**:
  + **Ukod:** Az Ügyfél egyed elsődleges kulcsa
  + **Adószám:** Az ügyfél adószáma
  + **Cim:** Összetetttulajdonság, mivel az ügyfél címe város, utca, házszámból áll
  + **Név:** Az ügyfél neve
  + **Email:** Email cím amin az ügyfél elérhető

 1. ábra: Vállalat ER modellje

## 1.2 Az adatbázis konvertálása XDM modellre

**XDM** modellnél háromféle jelölés használunk: ellipszist, rombuszt és téglalapot. Az ellipszis jelöli az elemeket, minden egyedből elem lesz, illetve a tulajdonságokból is. A rombusz jelöli az attribútumokat, amelyek a kulcs tulajdonságokból keletkeznek. A téglalap jelöli a szöveget, amely majd az **XML** dokumentumban fog megjelenni. Azoknak az elemeknek, amelyek többször is előfordulhatnak, a jelölése dupla ellipszissel történik. Az idegenkulcsok és a kulcsok közötti kapcsolatot szaggatott vonalas nyíllal jelöljük.

**XDM** **modell**:



2.ábra: Vállalat XDM modellje

# 1.3 Az XDM modell alapján XML dokumentum készítése

Az **XDM** modell alapján készítettem el az **XML** dokumentumot. Legelőször a gyökérelemmel kezdtem, amelynek a „**vallalat**” nevet adtam. Ezek után a gyermekelemeiből eltérő módon hoztam létre példányokat.

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

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

xsi:noNamespaceSchemaLocation="XMLSchemaY4O4X0.xsd">

<!--Részlegek-->

<reszleg rkod="r1">

<nev>Elektronika</nev>

<emelet>2</emelet>

<letszam>12</letszam>

<!--Csapatok-->

<csapat cskod="cs1" rkod="r1">

<vezeto>Szabó Ernő</vezeto>

<letszam>8</letszam>

<elorehaladas>30</elorehaladas>

<!--Dolgozok-->

<dolgozo dkod="d1">

<fizetes>3000000</fizetes>

<vegzettseg>Bsc diploma</vegzettseg>

<belepesiDatum>2006-02-02</belepesiDatum>

<beosztas bkod="cs1">

<munkaido>

<tol>10:00:00</tol>

<ig>18:00:00</ig>

</munkaido>

<pozicio>Szoftverfejlesztő</pozicio>

</beosztas>

<nev>Szabó Nórbert</nev>

</dolgozo>

<dolgozo dkod="d2">

<fizetes>200000</fizetes>

<vegzettseg>Középfokú végzettség</vegzettseg>

<belepesiDatum>2010-01-21</belepesiDatum>

<beosztas bkod="cs1">

<munkaido>

<tol>10:00:00</tol>

<ig>18:00:00</ig>

</munkaido>

<pozicio>Tesztelő</pozicio>

</beosztas>

<nev>Illés János</nev>

</dolgozo>

<dolgozo dkod="d3">

<fizetes>300000</fizetes>

<vegzettseg>Bsc diploma</vegzettseg>

<belepesiDatum>2000-03-03</belepesiDatum>

<beosztas bkod="cs1">

<munkaido>

<tol>10:00:00</tol>

<ig>18:00:00</ig>

</munkaido>

<pozicio>Tesztelő</pozicio>

</beosztas>

<nev>Fekete Csaba</nev>

</dolgozo>

</csapat>

<csapat cskod="cs2" rkod="r1">

<vezeto>Nagy István</vezeto>

<letszam>43</letszam>

<elorehaladas>75</elorehaladas>

<dolgozo dkod="d4">

<fizetes>242000</fizetes>

<vegzettseg>Bsc diploma</vegzettseg>

<belepesiDatum>2017-03-08</belepesiDatum>

<beosztas bkod="cs2">

<munkaido>

<tol>10:00:00</tol>

<ig>18:00:00</ig>

</munkaido>

<pozicio>Projekt menedzser</pozicio>

</beosztas>

<nev>Kiss Nikolett</nev>

</dolgozo>

</csapat>

<csapat cskod="cs3" rkod="r1">

<vezeto>Kovács András</vezeto>

<letszam>15</letszam>

<elorehaladas>45</elorehaladas>

<dolgozo dkod="d5">

<fizetes>3000000</fizetes>

<vegzettseg>Bsc diploma</vegzettseg>

<belepesiDatum>2011-11-10</belepesiDatum>

<beosztas bkod="cs3">

<munkaido>

<tol>10:00:00</tol>

<ig>18:00:00</ig>

</munkaido>

<pozicio>Szoftverfejlesztő</pozicio>

</beosztas>

<nev>Szabó Zsófia</nev>

</dolgozo>

</csapat>

<!--Projektek-->

<projekt pkod="p1" rkod="r1">

<nev>Fejlesztési projekt</nev>

<felelos>Szabó Ernő</felelos>

<hatarido>2024-02-02</hatarido>

</projekt>

<projekt pkod="p2" rkod="r1">

<nev>Karbantartási projekt</nev>

<felelos>Kovács András</felelos>

<hatarido>2024-05-15</hatarido>

</projekt>

<projekt pkod="p3" rkod="r1">

<nev>Tesztelési projekt</nev>

<felelos>Nagy István</felelos>

<hatarido>2024-03-30</hatarido>

</projekt>

<!--Ügyfelek-->

<ugyfel ukod="u1" pkod="p1">

<nev>János és Társai Kft.</nev>

<adoszam>12345678-1-12</adoszam>

<cim>

<varos>Miskolc</varos>

<utca>Petőfi út</utca>

<hazszam>22</hazszam>

</cim>

<email>janostarsai@gmail.com</email>

</ugyfel>

<ugyfel ukod="u2" pkod="p2">

<nev>Banános Joe kft</nev>

<adoszam>12573438-1-12</adoszam>

<cim>

<varos>Kistarcs</varos>

<utca>Arany János utca</utca>

<hazszam>8</hazszam>

</cim>

<email>bananajoe@gmail.com</email>

</ugyfel>

<ugyfel ukod="u3" pkod="p3">

<nev>Szepesi Kft.</nev>

<adoszam>22245678-1-12</adoszam>

<cim>

<varos>Eger</varos>

<utca>Puskás öcsi út</utca>

<hazszam>34</hazszam>

</cim>

<email>Szepesi@gmail.com</email>

</ugyfel>

</reszleg>

<!--Részleg-->

<reszleg rkod="r2">

<nev>Marketing</nev>

<emelet>3</emelet>

<letszam>10</letszam>

<!--Csapatok-->

<csapat cskod="cs4" rkod="r2">

<vezeto>Kiss Anna</vezeto>

<letszam>5</letszam>

<elorehaladas>60</elorehaladas>

<!--Dolgozok-->

<dolgozo dkod="d6">

<fizetes>4000000</fizetes>

<vegzettseg>Msc diploma</vegzettseg>

<belepesiDatum>2015-05-12</belepesiDatum>

<beosztas bkod="cs4">

<munkaido>

<tol>09:00:00</tol>

<ig>17:00:00</ig>

</munkaido>

<pozicio>Tesztelő</pozicio>

</beosztas>

<nev>Lakatos Tibor</nev>

</dolgozo>

<dolgozo dkod="d7">

<fizetes>3500000</fizetes>

<vegzettseg>Bsc diploma</vegzettseg>

<belepesiDatum>2018-03-04</belepesiDatum>

<beosztas bkod="cs4">

<munkaido>

<tol>09:00:00</tol>

<ig>17:00:00</ig>

</munkaido>

<pozicio>Szoftverfejlesztő</pozicio>

</beosztas>

<nev>Kunhalasi Ágnes</nev>

</dolgozo>

<dolgozo dkod="d8">

<fizetes>2500000</fizetes>

<vegzettseg>Bsc diploma</vegzettseg>

<belepesiDatum>2017-08-19</belepesiDatum>

<beosztas bkod="cs4">

<munkaido>

<tol>09:00:00</tol>

<ig>17:00:00</ig>

</munkaido>

<pozicio>Mérnök</pozicio>

</beosztas>

<nev>Fehér Zoltán</nev>

</dolgozo>

</csapat>

<csapat cskod="cs5" rkod="r2">

<vezeto>Nagy Zoltán</vezeto>

<letszam>5</letszam>

<elorehaladas>45</elorehaladas>

<dolgozo dkod="d9">

<fizetes>3000000</fizetes>

<vegzettseg>Bsc diploma</vegzettseg>

<belepesiDatum>2020-09-11</belepesiDatum>

<beosztas bkod="cs5">

<munkaido>

<tol>09:00:00</tol>

<ig>17:00:00</ig>

</munkaido>

<pozicio>Projekt menedzser</pozicio>

</beosztas>

<nev>Simon Gábor</nev>

</dolgozo>

</csapat>

<!--Projektek-->

<projekt pkod="p4" rkod="r2" >

<nev>Brand Építés</nev>

<felelos>Kiss Anna</felelos>

<hatarido>2024-03-15</hatarido>

</projekt>

<projekt pkod="p5" rkod="r2" >

<nev>SEO Optimalizálás</nev>

<felelos>Nagy Zoltán</felelos>

<hatarido>2024-04-20</hatarido>

</projekt>

<!--Ügyfelek-->

<ugyfel ukod="u4" pkod="p4">

<nev>Kolompár Kft.</nev>

<adoszam>87654321-1-12</adoszam>

<cim>

<varos>Debrecen</varos>

<utca>Fő utca</utca>

<hazszam>10</hazszam>

</cim>

<email>Kolompar@domain.com</email>

</ugyfel>

<ugyfel ukod="u5" pkod="p5">

<nev>Csizmadia Kft.</nev>

<adoszam>81154321-1-12</adoszam>

<cim>

<varos>Kőszeghalom</varos>

<utca>Nyircsák utca</utca>

<hazszam>19</hazszam>

</cim>

<email>Csiz@domain.com</email>

</ugyfel>

</reszleg>

<reszleg rkod="r3">

<nev>HR</nev>

<emelet>2</emelet>

<letszam>6</letszam>

<!--Csapatok-->

<csapat cskod="cs6" rkod="r3">

<vezeto>Tóth Péter</vezeto>

<letszam>3</letszam>

<elorehaladas>80</elorehaladas>

<!--Dolgozók-->

<dolgozo dkod="d10">

<fizetes>5000000</fizetes>

<vegzettseg>Msc diploma</vegzettseg>

<belepesiDatum>2012-10-10</belepesiDatum>

<beosztas bkod="cs6">

<munkaido>

<tol>09:00:00</tol>

<ig>17:00:00</ig>

</munkaido>

<pozicio>Projekt menedzser</pozicio>

</beosztas>

<nev>Juhász Péter</nev>

</dolgozo>

<dolgozo dkod="d11">

<fizetes>3500000</fizetes>

<vegzettseg>Bsc diploma</vegzettseg>

<belepesiDatum>2015-05-22</belepesiDatum>

<beosztas bkod="cs6">

<munkaido>

<tol>09:00:00</tol>

<ig>17:00:00</ig>

</munkaido>

<pozicio>Mérnök</pozicio>

</beosztas>

<nev>Tabácsi János</nev>

</dolgozo>

<dolgozo dkod="d12">

<fizetes>3000000</fizetes>

<vegzettseg>Bsc diploma</vegzettseg>

<belepesiDatum>2018-07-05</belepesiDatum>

<beosztas bkod="cs6">

<munkaido>

<tol>09:00:00</tol>

<ig>17:00:00</ig>

</munkaido>

<pozicio>Mérnök</pozicio>

</beosztas>

<nev>Orbán Kata</nev>

</dolgozo>

</csapat>

<csapat cskod="cs7" rkod="r3">

<vezeto>Varga Eszter</vezeto>

<letszam>3</letszam>

<elorehaladas>70</elorehaladas>

<dolgozo dkod="d13">

<fizetes>2500000</fizetes>

<vegzettseg>Bsc diploma</vegzettseg>

<belepesiDatum>2019-11-14</belepesiDatum>

<beosztas bkod="cs7">

<munkaido>

<tol>09:00:00</tol>

<ig>17:00:00</ig>

</munkaido>

<pozicio>Aszisztens</pozicio>

</beosztas>

<nev>Tisza Szilvia</nev>

</dolgozo>

</csapat>

<!--Projektek-->

<projekt pkod="p6" rkod="r3" >

<nev>Alkalmazotti elégedettség felmérés</nev>

<felelos>Tóth Péter</felelos>

<hatarido>2024-06-30</hatarido>

</projekt>

<projekt pkod="p7" rkod="r3" >

<nev>Jutalom és elismerés program</nev>

<felelos>Varga Eszter</felelos>

<hatarido>2024-08-15</hatarido>

</projekt>

<!--Ügyfelek-->

<ugyfel ukod="u6" pkod="p6">

<nev>ABC Tanácsadó Kft.</nev>

<adoszam>11223344-1-12</adoszam>

<cim>

<varos>Szeged</varos>

<utca>Kossuth Lajos utca</utca>

<hazszam>15</hazszam>

</cim>

<email>abctanacsado@gmail.com</email>

</ugyfel>

<ugyfel ukod="u7" pkod="p7">

<nev>aliKebab Kft</nev>

<adoszam>17773344-1-12</adoszam>

<cim>

<varos>Pozsony</varos>

<utca>Kiss Ferencz utca</utca>

<hazszam>5</hazszam>

</cim>

<email>aAlikebab@gmail.com</email>

</ugyfel>

</reszleg>

## </vallalat>

## 1.4 Az XML dokumentum alapján XMLSchema készítése

Az **XML** dokumentumhoz készítettem egy validációt elősegítő **XML** séma definíciót (**XSD**). Először létrehoztam az egyszerű típusokat, amelyek különböző adatmezők típusát és korlátozásait határozzák meg. Például az email címhez reguláris kifejezéssel meghatároztam egy mintát, amely biztosítja a helyes formátumot. Az **adoszam** típus is egy reguláris kifejezést tartalmaz, míg a **pozicio** esetében enumerációval korlátoztam a lehetséges értékeket.

Ezután elkészítettem a komplex típusokat minden főbb elemre, mint például a **Csapat**, **Reszleg**, **Projekt**, **Dolgozo** és **Ugyfel**, amelyek különböző al-elemekből állnak és az üzleti struktúra főbb egységeit képviselik. Minden elemhez meghatároztam a szükséges attribútumokat is, mint például az egyedi azonosítókat (**ukod**, **rkod**, **pkod**, stb.), amelyek egyértelműen azonosítják az egyes entitásokat.

A séma elsődleges és idegen kulcsokat is tartalmaz, amelyek biztosítják az adatok integritását és az egyes entitások közötti kapcsolatokat, például a **Csapat** és **Reszleg** közötti hivatkozásokat. Így biztosítottam, hogy a **Csapat** mindig egy konkrét **Reszleg**-hez tartozzon, és az **Ugyfel**-ek is az egyes **Projekt**-ekhez legyenek rendelve.

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

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<!-- Elemi jelölőelemek definiálása -->

<xs:element name="csapat" type="csapatTipus" />

<xs:element name="projekt" type="projektTipus" />

<xs:element name="ugyfel" type="ugyfelTipus" />

<xs:element name="dolgozo" type="dolgozoTipus" />

<!--Email tipus létrehozása, korlátozása reguláris kifejezéssel-->

<xs:simpleType name="emailTipus">

<xs:restriction base="xs:string">

<!--regex az adószámra:

[0-9]{8}: Pontosan 8 számjegy.

-: Egy kötőjel.

[0-9]{1}: Pontosan 1 számjegy.

-: Egy kötőjel.

[0-9]{2}: Pontosan 2 számjegy.-->

<xs:pattern value="[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}"></xs:pattern>

</xs:restriction>

</xs:simpleType>

<!--Adoszam tipus létrehozása, korlátozása reguláris kifejezéssel-->

<xs:simpleType name="adoszamTipus">

<xs:restriction base="xs:string">

<!--regex az adószámra:

[0-9]{8}: Pontosan 8 számjegy.

-: Egy kötőjel.

[0-9]{1}: Pontosan 1 számjegy.

-: Egy kötőjel.

[0-9]{2}: Pontosan 2 számjegy.-->

<xs:pattern value="[0-9]{8}-[0-9]{1}-[0-9]{2}"></xs:pattern>

</xs:restriction>

</xs:simpleType>

<!--Pozício tipus létrehozása, korlátozása adott értékekre-->

<xs:simpleType name="pozicioTipus">

<xs:restriction base="xs:string">

<xs:enumeration value="Szoftverfejlesztő"></xs:enumeration>

<xs:enumeration value="Projekt menedzser"></xs:enumeration>

<xs:enumeration value="Mérnök"></xs:enumeration>

<xs:enumeration value="Aszisztens"></xs:enumeration>

<xs:enumeration value="Tesztelő"></xs:enumeration>

</xs:restriction>

</xs:simpleType>

<!--Előrehaladás tipus létrehozása, korlátozása adott intervallumra-->

<xs:simpleType name="elorehaladasTipus">

<xs:restriction base="xs:int">

<xs:minExclusive value="0"></xs:minExclusive>

<xs:maxExclusive value="100"></xs:maxExclusive>

</xs:restriction>

</xs:simpleType>

<!--Emelet tipus létrehozása, korlátozása adott értékekre-->

<xs:simpleType name="emeletTipus">

<xs:restriction base="xs:int">

<xs:minExclusive value="1"></xs:minExclusive>

<xs:maxExclusive value="6"></xs:maxExclusive>

</xs:restriction>

</xs:simpleType>

<!--Cím tipus létrehozása, gyerekelemek definilása-->

<xs:complexType name="cimTipus">

<xs:sequence>

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

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

<xs:element name="hazszam" type="xs:int"></xs:element>

</xs:sequence>

</xs:complexType>

<!--Ügyfél tipus létrehozása, gyerekelemek definilása-->

<xs:complexType name="ugyfelTipus">

<xs:sequence>

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

<xs:element name="adoszam" type="adoszamTipus"></xs:element>

<xs:element name="cim" type="cimTipus"></xs:element>

<xs:element name="email" type="emailTipus"></xs:element>

</xs:sequence>

<xs:attribute name="ukod" type="xs:string" use="required"></xs:attribute>

<xs:attribute name="pkod" type="xs:string" use="required"></xs:attribute>

</xs:complexType>

<!--Munkaidő tipus létrehozása, gyerekelemek definilása-->

<xs:complexType name="munkaidoTipus">

<xs:sequence>

<xs:element name="tol" type="xs:time"></xs:element>

<xs:element name="ig" type="xs:time"></xs:element>

</xs:sequence>

</xs:complexType>

<!--Projekt tipus létrehozása, gyerekelemek és attribútumai definilása-->

<xs:complexType name="projektTipus" mixed="true">

<xs:sequence>

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

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

<xs:element name="hatarido" type="xs:date"></xs:element>

</xs:sequence>

<xs:attribute name="pkod" type="xs:string" use="required"></xs:attribute>

<xs:attribute name="rkod" type="xs:string" use="required"></xs:attribute>

</xs:complexType>

<!--Beosztás tipus létrehozása, gyerekelemek és attribútumai definilása-->

<xs:complexType name="beosztasTipus">

<xs:sequence>

<xs:element name="munkaido" type="munkaidoTipus"></xs:element>

<xs:element name="pozicio" type="pozicioTipus"></xs:element>

</xs:sequence>

<xs:attribute name="bkod" type="xs:string" use="required"></xs:attribute>

</xs:complexType>

<!--Dolgozo tipus létrehozása, gyerekelemek és attribútumai definilása-->

<xs:complexType name="dolgozoTipus" mixed="true">

<xs:sequence>

<xs:element name="fizetes" type="xs:int"></xs:element>

<xs:element name="vegzettseg" type="xs:string" minOccurs="1" maxOccurs="unbounded"></xs:element>

<xs:element name="belepesiDatum" type="xs:date"></xs:element>

<xs:element name="beosztas" type="beosztasTipus"></xs:element>

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

</xs:sequence>

<xs:attribute name="dkod" type="xs:string" use="required"></xs:attribute>

</xs:complexType>

<!--Csapat tipus létrehozása, gyerekelemek és attribútumai definilása-->

<xs:complexType name="csapatTipus" mixed="true">

<xs:sequence>

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

<xs:element name="letszam" type="xs:int"></xs:element>

<xs:element name="elorehaladas" type="elorehaladasTipus"></xs:element>

<xs:element name="dolgozo" type="dolgozoTipus" minOccurs="1" maxOccurs="unbounded"></xs:element>

</xs:sequence>

<xs:attribute name="cskod" type="xs:string" use="required"></xs:attribute>

<xs:attribute name="rkod" type="xs:string" use="required"></xs:attribute>

</xs:complexType>

<!--Részleg egyed létrehozása, gyerekelemek és attribútumai definilása-->

<xs:element name="reszleg">

<xs:complexType mixed="true">

<xs:sequence>

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

<xs:element name="emelet" type="emeletTipus"></xs:element>

<xs:element name="letszam" type="xs:int"></xs:element>

<!-- Hivatkozások globális elemekre -->

<xs:element ref="csapat" minOccurs="1" maxOccurs="unbounded"></xs:element>

<xs:element ref="projekt" minOccurs="1" maxOccurs="unbounded"></xs:element>

<xs:element ref="ugyfel" minOccurs="1" maxOccurs="unbounded"></xs:element>

</xs:sequence>

<xs:attribute name="rkod" type="xs:string" use="required"></xs:attribute>

</xs:complexType>

</xs:element>

<!--Vállalat gyökérelem létrehozása (globális), gyerekelemek definiálása-->

<xs:element name="vallalat">

<xs:complexType>

<xs:sequence>

<xs:element ref="reszleg" minOccurs="1" maxOccurs="unbounded"></xs:element>

</xs:sequence>

</xs:complexType>

<!--Elsődleges kulcsok-->

<xs:key name="reszleg\_kulcs">

<xs:selector xpath="reszleg"></xs:selector>

<xs:field xpath="@rkod"></xs:field>

</xs:key>

<xs:key name="csapat\_kulcs">

<xs:selector xpath="reszleg/csapat"></xs:selector>

<xs:field xpath="@cskod"></xs:field>

</xs:key>

<xs:key name="projekt\_kulcs">

<xs:selector xpath="reszleg/projekt"></xs:selector>

<xs:field xpath="@pkod"></xs:field>

</xs:key>

<xs:key name="ugyfel\_kulcs">

<xs:selector xpath="reszleg/ugyfel"></xs:selector>

<xs:field xpath="@ukod"></xs:field>

</xs:key>

<xs:key name="dolgozo\_kulcs">

<xs:selector xpath="reszleg/csapat/dolgozo"></xs:selector>

<xs:field xpath="@dkod"></xs:field>

</xs:key>

<!--Idegen kulcsok-->

<xs:keyref name="csapat\_tartozik\_reszleg\_kulcs" refer="reszleg\_kulcs">

<xs:selector xpath="csapat"></xs:selector>

<xs:field xpath="@rkod"></xs:field>

</xs:keyref>

<xs:keyref name="projekt\_tartozik\_reszleg\_kulcs" refer="reszleg\_kulcs">

<xs:selector xpath="projekt"></xs:selector>

<xs:field xpath="@rkod"></xs:field>

</xs:keyref>

<xs:keyref name="ugyfel\_rendel\_projekt\_kulcs" refer="projekt\_kulcs">

<xs:selector xpath="ugyfel"></xs:selector>

<xs:field xpath="@pkod"></xs:field>

</xs:keyref>

<xs:keyref name="dolgozo\_tartozik\_csapat\_kulcs" refer="csapat\_kulcs">

<xs:selector xpath="csapat/dolgozo"></xs:selector>

<xs:field xpath="@cskod"></xs:field>

</xs:keyref>

<!-- Egyediség az 1:1 kapcsolat biztosítására

Nincs szükség unique-ra: Az ugyfel és projekt elemek külön kulcsai (ukod és pkod) már egyedi azonosítóként funkcionálnak.

<xs:unique name="ugyfel\_es\_projekt\_egyedi">

<xs:selector xpath="reszleg/ugyfel | reszleg/projekt"></xs:selector>

<xs:field xpath="@pkod"></xs:field>

</xs:unique>

-->

</xs:element>

# </xs:schema>

# 2. DOM program készítése JAVA környezetbe

A **DOM** programokat **JAVA**-ban készítettem el, ahogy a feladatkiírásban is szerepelt. Az alábbi programokat a következő alfejezetekben fogom részletesebben taglalni.

## 2.1 DOM adatolvasás

**DOMReadY4O4X0**:  
Ez a fő metódus, amely az **XML** fájlt olvassa be és dolgozza fel **DOM** (Document Object Model) módszerrel. Betölti az **XML** dokumentumot, kiírja a gyökérelemet, majd végigiterál a "**reszleg**", "**csapat**", "**dolgozo**", "**projekt**" és "**ugyfel**" elemek között, kiírva azok információit. A **printToFile** metódust is meghívja, hogy az átalakított dokumentumot egy új fájlba mentse.

**printToFile**:  
Ez a metódus az **XML** dokumentumot egy új fájlba (**XMLY4O4X02**.**xml**) írja ki. A **Transformer** osztály segítségével az **XML** dokumentumot formázott szöveggé alakítja, majd elmenti a megadott helyre.

**buildDocument**:  
Ez a metódus egy fájl elérési utat vár bemenetként, létrehoz egy **DocumentBuilder**-t, és az adott **XML** fájlt egy **Document** objektummá alakítja. A dokumentumot normalizálja, hogy biztosítsa a megfelelő struktúrát, majd visszaadja a dokumentumot.

**kiirReszlegInfo**:  
Ez a metódus a "reszleg" (**részleg**) elemről ír ki információkat, például a részleg kódját (**rkod**), nevét (**nev**), emeletét (**emelet**) és létszámát (**letszam**). Az adatokat az **XML** dokumentumból olvassa ki és a konzolra írja.

**kiirCsapatInfo**:  
Ez a metódus a "csapat" elemről ír ki információkat, például a csapat kódját (**cskod**), a részleg kódját (**rkod**), a vezetőt (**vezeto**) és az előrehaladást (**elorehaladas**). Az adatokat az **XML** elemből szerzi be és jeleníti meg.

**kiirDolgozoInfo**:  
Ez a metódus a "**dolgozo**" (dolgozó) elemről ad ki információkat, például a dolgozó kódját (**dkod**), fizetését (**fizetes**), végzettségét (**vegzettseg**), belépési dátumát (**belepesiDatum**) és nevét (**nev**). Meghívja a **kiirBeosztasInfo** metódust, hogy további adatokat jelenítsen meg a dolgozó beosztásáról.

**kiirBeosztasInfo**:  
Ez a metódus a dolgozó beosztásáról ír ki információkat, például a beosztás kódját (**bkod**), a pozíciót (**pozicio**) és a munkaidőt. A munkaidő adatokat a kiirMunkaidoInfo metódus segítségével jeleníti meg.

**kiirMunkaidoInfo**:  
Ez a metódus a munkaidővel kapcsolatos adatokat írja ki, például a munkaidő kezdetét (**tol**) és végét (**ig**). Az adatokat az **XML** elem alapján olvassa ki és jeleníti meg.

**kiirProjektInfo**:  
Ez a metódus a "**projekt**" elemről ír ki információkat, például a projekt kódját (**pkod**), a részleg kódját (**rkod**), és az ügyfél kódját (**ukod**). További adatokat is kiír a projekt nevéről (**nev**), a felelős személyről (**felelos**) és a határidőről (**hatarido**).

**kiirUgyfelInfo**:  
Ez a metódus az "**ugyfel**" (ügyfél) elemről ír ki információkat, például az ügyfél kódját (**ukod**), a részleg kódját (**rkod**), a nevét (**nev**), adószámát (**adoszam**), címét és email címét (**email**). A cím adatait a **kiirUgyfelCimInfo** metódus segítségével írja ki.

**kiirUgyfelCimInfo**:  
Ez a metódus az ügyfél címével kapcsolatos adatokat írja ki, például a várost (**varos**), utcát (**utca**) és házszámot (**hazszam**). Az **XML** elem alapján olvassa ki és jeleníti meg az adatokat.

**Fontosnak** **tartott** **kódrészlet elérhető itt:** [[GitHub link a konkrét kódrészlethez](https://github.com/szaloczy/Y4O4X0WebXML/blob/95213ffab229ada776dff71daa84f734909aff81/XMLTaskY4O4X0/DOMParseY4O4X0/src/hu/domparse/y4o4x0/DOMReadY4O4X0.java#L15)]

public static void main(String[] args) {

try {

// Az XML fájl elérési útjának megadása

File xmlFile = new File("src/resources/XMLY4O4X0.xml");

// Az XML dokumentum felépítése (DOM objektum létrehozása)

Document doc = buildDocument(xmlFile);

// A gyökér elem kiírása

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

// A "reszleg" elemek listájának lekérése

NodeList reszlegList = doc.getDocumentElement().getElementsByTagName("reszleg");

// Végighalad a "reszleg" elemek listáján

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

Node reszlegNode = reszlegList.item(i);

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

System.out.println("\n\tJelenlegi elem: " + reszlegNode.getNodeName());

//Tipus ellenőrzés

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

Element reszleg = (Element) reszlegNode;

// Részleg információinak kiírása

kiirReszlegInfo(reszleg);

// "csapat" elemek lekérése és feldolgozása

NodeList csapatList = reszleg.getElementsByTagName("csapat");

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

Node csapatNode = csapatList.item(j);

System.out.println("\n\tJelenlegi elem: " + csapatNode.getNodeName());

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

Element csapat = (Element) csapatNode;

// Csapat információinak kiírása

kiirCsapatInfo(csapat);

// "dolgozo" elemek lekérése és feldolgozása

NodeList dolgozoList = reszleg.getElementsByTagName("dolgozo");

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

Node dolgozoNode = dolgozoList.item(k);

System.out.println("\n\tJelenlegi elem: " + dolgozoNode.getNodeName());

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

Element dolgozo = (Element) dolgozoNode;

kiirDolgozoInfo(dolgozo);

}

}

}

}

// "projekt" elemek lekérése és feldolgozása

NodeList projektLista = reszleg.getElementsByTagName("projekt");

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

Node projektNode = projektLista.item(j);

System.out.println("\n\t jelenlegi elem: " + projektNode.getNodeName());

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

Element projekt = (Element) projektNode;

// Projekt információinak kiírása

kiirProjektInfo(projekt);

}

}

// "ugyfel" elemek lekérése és feldolgozása

NodeList ugyfelList = reszleg.getElementsByTagName("ugyfel");

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

Node ugyfelNode = ugyfelList.item(0);

System.out.println("\n\t Jelenlegi elem: " + ugyfelNode.getNodeName());

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

Element ugyfel = (Element) ugyfelNode;

// Ügyfél információinak kiírása

kiirUgyfelInfo(ugyfel);

}

}

}

}

printToFile(doc);

} catch (Exception e) {

System.out.println(e.getMessage());

}

}

**Main metódus:**

A kódban **XML** dokumentumot dolgozunk fel, amelyben több egymásba ágyazott elem található, mint például a "**reszleg**", "**csapat**", "**dolgozo**", "**projekt**" és "**ugyfel**". A kód első lépésben lekéri a "**reszleg**" elemeket, majd minden egyes "**reszleg**" elemhez hozzáférve kiírja annak információit. Ezt követően, a "**reszleg**" elemeken belül tovább keres, és a "**csapat**", "**dolgozo**", "**projekt**", illetve "**ugyfel**" elemeket is feldolgozza.

Minden egyes elemhez külön-külön funkciókat használ (pl. **kiirReszlegInfo**, **kiirCsapatInfo**), hogy az adott elemhez tartozó információkat kiírja a konzolra. A "dolgozo" és "projekt" elemek esetén hasonló lépések történnek, mint a "**csapat**" esetében.

Fontos, hogy az elem típusát is ellenőrzi, hogy biztosan csak a megfelelő típusú (**ELEMENT**\_**NODE**) elemeket dolgozza fel, és minden szinten átvizsgálja az összes megfelelő gyermekelemet. A kód minden egyes elem feldolgozása előtt kiírja annak nevét a konzolra.

## 2.2 DOM adatírás

**DOMWriteY4O4X0**:  
Ez a Java osztály **XML** dokumentumot hoz létre, és kiírja a konzolra. A program az **XML** elemeket **DOM** (Document Object Model) segítségével manipulálja.

**main**:  
A fő metódus, amely elindítja a programot. Meghívja a **createAndPrintXMLDocument** metódust, és kezeli az esetleges hibákat.

**createAndPrintXMLDocument**:  
Létrehozza az **XML** dokumentumot, beállítja a gyökérelemet (**namespace**-ekkel), és hozzáadja a részlegeket. A dokumentumot később kiírja a konzolra a **printDocumentToConsole** metódus segítségével.

**addDepartment**:  
Hozzáad egy "reszleg" (**részleg**) elemet a dokumentumhoz. A részleghez tartozó információkat (**kód**, **név**, **emelet**, **létszám**) és a hozzá tartozó csapatokat, projekteket és ügyfeleket adja hozzá.

**addTeam**:  
Hozzáad egy "**csapat**" elemet a részleghez, amely tartalmazza a csapat kódját, vezetőt, létszámot és előrehaladást. Emellett hozzáadja a csapathoz a dolgozókat is.

**addEmployee**:  
Hozzáad egy "dolgozo" (**dolgozó**) elemet a csapathoz. Az elem tartalmazza a dolgozó kódját, fizetését, végzettségét, belépési dátumát, munkaidő adatait, pozícióját és nevét. A munkaidő és beosztás részleteit is hozzáadja a dolgozóhoz.

**addProject**:  
Hozzáad egy "**projekt**" elemet a részleghez, amely tartalmazza a projekt kódját, nevét, felelős vezetőt és határidőt.

**addClient**:  
Hozzáad egy "**ugyfel**" (ügyfél) elemet a részleghez. Az elem tartalmazza az ügyfél kódját, nevét, adószámát, címét és email címét.

**appendElement**:  
Ez a segédmetódus egy új **XML** elemet ad hozzá a szülő elemhez. Az elem tartalmazza a megadott címkét (**tagName**) és szöveges tartalmat (**textContent**).

**printDocumentToConsole**:  
Ez a metódus kiírja az **XML** dokumentumot a konzolra formázott módon. A **Transformer** osztály segítségével történik a kiírás, amely biztosítja az **XML** dokumentum megfelelő formázását.

**Fontosnak** **tartott** **kódrészlet elérhető itt:** [[GitHub link a konkrét kódrészlethez](https://github.com/szaloczy/Y4O4X0WebXML/blob/95213ffab229ada776dff71daa84f734909aff81/XMLTaskY4O4X0/DOMParseY4O4X0/src/hu/domparse/y4o4x0/DOMWriteY4O4X0.java#L88)]

//Dolgozó Hozzáadása

private static void addEmployee(Document document, Element teamElement, String dskod, String fizetes, String vegzettseg, String szuletes, String munkaido\_kezd, String munkaido\_veg, String pozicio, String nev) {

Element employee = document.createElement("dolgozo");

employee.setAttribute("dkod", dskod);

//gyerekelem hozzáfűzés

appendElement(document, employee, "fizetes", fizetes);

appendElement(document, employee, "vegzettseg", vegzettseg);

appendElement(document, employee, "belepesiDatum", szuletes);

//gyerekelem hozzáfűzés

Element workTime = document.createElement("munkaido");

appendElement(document, workTime, "tol", munkaido\_kezd);

appendElement(document, workTime, "ig", munkaido\_veg);

employee.appendChild(workTime);

//gyerekelem hozzáfűzés

Element position = document.createElement("beosztas");

position.setAttribute("bkod", "b1"); // Igazítsuk a tényleges logikához

appendElement(document, position, "pozicio", pozicio);

employee.appendChild(position);

appendElement(document, employee, "nev", nev);

teamElement.appendChild(employee);

}

**AddEmployee metódus:**

A **addEmployee** metódus célja, hogy egy új "**dolgozo**" (dolgozó) elemet adjon hozzá a csapathoz az **XML** dokumentumban. A metódus létrehozza a "**dolgozo**" elemet, és beállítja a dolgozó kódját (**dkod**), amit a paramétereként kap. Ezt követően hozzáadja a dolgozó fizetését, végzettségét és belépési dátumát új **XML** elemek formájában. A munkaidőt egy "**munkaido**" elembe ágyazza, amely tartalmazza a kezdési és befejezési időpontot. Ezen kívül hozzáadja a dolgozó pozícióját is egy "**beosztas**" elem segítségével, melynek tartalmaznia kell a pozíció nevét és egy egyedi kódot. Végül a dolgozó nevét is beilleszti az "**nev**" elemben, és az összes adatot a csapathoz rendeli, befejezve a dolgozó adatainak **XML** struktúrába illesztését. A metódus tehát egy dolgozó teljes adatainak hozzáadását végzi el az **XML** dokumentumban, figyelembe véve az összes szükséges részletet.

## 2.3 DOM adatlekérdezés

**main**

A **main** metódus felelős az **XML** fájl beolvasásáért és az egyes lekérdező metódusok meghívásáért. Az **XML** fájl elérési útját beállítja, létrehozza a **DOM** dokumentumot, majd a **queryForAllTeamAndLeader**,**employeesWhoseSalaryOverAMillion**,**projectDeadlinesBefore2024** és **allCustomersNameAndEmail** metódusokat hívja meg az adatok lekérdezésére és kiírására.

**allCustomersNameAndEmail**

Ez a metódus az **XML** dokumentumból lekérdezi az összes ügyfél nevét és email címét. A **NodeList** segítségével végigiterálja az "**ugyfel**" elemeket, és minden egyes ügyfélnél kiírja a nevét és email címét.

**queryForAllTeamAndLeader**

A **queryForAllTeamAndLeader** metódus az **XML** fájlban található csapatokat és azok vezetőit kérdezi le. Végigiterál az összes "**csapat**" elemeken, és minden csapat esetében kiírja annak kódját és vezetőjének nevét.

**employeesWhoseSalaryOverAMillion**

Ez a metódus a dolgozók között keres, és csak azokat írja ki, akiknek a fizetése meghaladja az 1 millió forintot. Az **XML**-ből lekérdezi a dolgozók nevét és fizetését, majd összehasonlítja a fizetést a küszöbértékkel (1 millió forint).

**projectDeadlinesBefore2024**

A **projectDeadlinesBefore2024** metódus az **XML**-ben található projektek határidejét vizsgálja meg, és azokat a projekteket írja ki, amelyek határideje 2024. május 1-je előtt van. A projektek "**hatarido**" elemét hasonlítja össze a kívánt dátummal, és ha a projekt határideje korábbi, kiírja a projekt nevét és határidejét.

**Fontosnak** **tartott** **kódrészlet elérhető itt:** [[GitHub link a konkrét kódrészlethez](https://github.com/szaloczy/Y4O4X0WebXML/blob/8b12c865fdee6a92ef159c383f5095783941306d/XMLTaskY4O4X0/DOMParseY4O4X0/src/hu/domparse/y4o4x0/DOMQueryY4O4X0.java#L82)]

public static void projectDeadlinesBefore2024(Document doc) {

System.out.println("\n3. Projektek határideje 2024 május előtt:");

NodeList projektek = doc.getElementsByTagName("projekt");

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

Element projekt = (Element) projektek.item(i);

String hatarido = projekt.getElementsByTagName("hatarido").item(0).getTextContent();

if (hatarido.compareTo("2024-05-01") < 0) {

String nev = projekt.getElementsByTagName("nev").item(0).getTextContent();

System.out.println(" - Projekt név: " + nev + ", Határidő: " + hatarido);

}

}

}

**ProjectDeadlinesBefore2024 metódus**

A **projectDeadlinesBefore2024** metódus az **XML** dokumentumban található projektek határidejét ellenőrzi, és csak azokat a projekteket írja ki, amelyek határideje 2024. május 1. előtt van. Az **XML**-ben található "**projekt**" elemeket vizsgálja, és minden projekt "**hatarido**" elemét lekérdezi. Ha a projekt határideje korábbi, mint a megadott dátum, akkor kiírja a projekt nevét és a határidőt. A metódus így segít azonosítani a közelgő határidejű projekteket. Az összes projektet végignézve a metódus azokkal a projektek adataival szolgál, amelyek megfelelnek a dátumfeltételnek.

## 2.4 DOM adatmódosítás

**DOMModifyY4O4X0**: Ez az osztály különböző műveleteket végez egy **XML** fájlon. A **main** metódus beolvassa a fájlt, majd meghívja a módosító metódusokat, amelyek frissítik a dolgozók fizetését, új projekteket adnak a marketing részleghez, módosítják egy ügyfél e-mail címét, és növelik a csapat létszámát. A végén a **printModifiedXml** metódus kiírja a módosított fájlt a konzolra. Az osztály az **XML** fájl struktúráját az alkalmazás igényei szerint módosítja.

**increaseNOTeamMembers**: Ez a metódus a csapatok számát növeli, konkrétan a "**cs2**" kódú csapat létszámát növeli meg 100 fővel. A **csapat** elemeket végignézve az rkod és cskod attribútumok alapján találja meg a megfelelő csapatot, majd módosítja a letszam elem értékét.

**updateACostumerEmailAddress**: Ez a metódus egy adott ügyfél e-mail címét módosítja. Az **ugyfel** elemek között keres egy olyan ügyfelet, akinek az ukod attribútuma "**u1**", majd frissíti az e-mail címét egy új értékre ("**ujemail**@**domain**.**com**").

**addNewProjectToMarketingDepartment**: A metódus egy új projektet ad hozzá a marketing részleghez. Az **rkod** attribútum alapján keresi meg a marketing részleget, majd új projektet hoz létre, amely tartalmazza a projekt nevét, felelősét és határidejét. A projekt a megfelelő **reszleg** elemhez kerül hozzáadásra.

**printModifiedXml**: Ez a metódus a módosított **XML** dokumentumot kiírja a konzolra. A **Transformer** osztály segítségével az **XML** dokumentumot jól olvasható formátumban jeleníti meg, behúzásokkal, hogy könnyen áttekinthető legyen.

**modifyEmployeeSalary**: A metódus a dolgozók fizetését módosítja. Az **XML**-ben szereplő dolgozo elemek között keres egy olyan dolgozót, akinek a dkod attribútuma "**d1**", majd frissíti annak fizetését az új értékre ("**500000**").

**Fontosnak** **tartott** **kódrészlet elérhető itt:** [[GitHub link a konkrét kódrészlethez](https://github.com/szaloczy/Y4O4X0WebXML/blob/8b12c865fdee6a92ef159c383f5095783941306d/XMLTaskY4O4X0/DOMParseY4O4X0/src/hu/domparse/y4o4x0/DOMModifyY4O4X0.java#L71)]

private static void addNewProjectToMarketingDepartment(Document doc) {

// A "reszleg" elemek listájának lekérése

NodeList reszlegek = doc.getElementsByTagName("reszleg");

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

Element reszleg = (Element) reszlegek.item(i);

if (reszleg.getAttribute("rkod").equals("r2")) {

// Új "projekt" létrehozása

Element ujProjekt = doc.createElement("projekt");

ujProjekt.setAttribute("pkod", "p10");

ujProjekt.setAttribute("rkod", "r2");

ujProjekt.setAttribute("ukod", "u6");

Element nev = doc.createElement("nev");

nev.setTextContent("XML beadandó készítés");

ujProjekt.appendChild(nev);

Element felelos = doc.createElement("felelos");

felelos.setTextContent("Szalóczy Krisztián");

ujProjekt.appendChild(felelos);

Element hatarido = doc.createElement("hatarido");

hatarido.setTextContent("2024-12-10");

ujProjekt.appendChild(hatarido);

// Az utolsó "projekt" után helyezés

NodeList projektek = reszleg.getElementsByTagName("projekt");

if (projektek.getLength() > 0) {

Element utolsoProjekt = (Element) projektek.item(projektek.getLength() - 1);

reszleg.insertBefore(ujProjekt, utolsoProjekt.getNextSibling());

} else {

// Ha nincs más projekt, a reszleg gyökérhez adja

reszleg.appendChild(ujProjekt);

}

}

}

}

**AddNewProjectToMarketingDepartment metódus**

A **addNewProjectToMarketingDepartment** metódus a marketing részleghez (**azonosítója**: "**r2**") új projektet ad hozzá az **XML** dokumentumban. Először lekéri az összes **reszleg** elemet, majd iterál azokon, hogy megtalálja azt, amelyik a marketing részleget képviseli az **rkod** attribútum "**r2**" értékével. Ha sikerül azonosítani a marketing részleget, a metódus létrehoz egy új projekt elemet, és beállítja annak a projekt kódját ("**p10**"), a részleg kódját ("**r2**") és egy ügyfél kódot ("**u6**"). Ezután három új gyerek elemet hoz létre: a projekt nevét ("**XML** **beadandó** **készítés**"), a projekt felelősét ("**Szalóczy** **Krisztián**") és a projekt határidejét ("**2024**-**12**-**10**"), majd ezeket hozzáadja a projekt elemhez. Végül az új projekt elemet hozzáfűzi a marketing részleghez, így az a dokumentumban a megfelelő helyen jelenik meg, és az **XML** struktúra frissül.