

A decorative graphic consisting of a thin gold circle on the left side. A thick black left square bracket is positioned vertically, overlapping the circle and a horizontal bar. The horizontal bar is divided into a gold-colored left portion and a white right portion. A thick gold right square bracket is positioned vertically on the right side of the white portion of the bar.

# Adatkezelés - XML

XML – 3 – XML séma

# [ XML séma (XSD) ]

- XML dokumentumok tartalmát és szerkezetét.
  - Elemeket, jellemzőket
  - Adat típusokat
  - Elemek számára adhatunk előírást
  - Elrendezést
- ~ DTD, de több annál
  - XML alapú
  - Lehetőséget rejt magában

# XSD példa

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="konyv">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="szerzo" type="xs:string"/>
        <xs:element name="cim" type="xs:string"/>
        <xs:element name="ar" type="xs:integer"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

## Az XML dokumentumban így kell megadni:

```
<?xml version="1.0" encoding="UTF-8"?>
<konyv xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="konyv.xsd">
  <szerzo>XY</szerzo>
  <cim>XML tananyag</cim>
  <ar>1000</ar>
</konyv>
```

# [ XSD és DTD ]

## XML:

```
<?xml version="1.0"?>
<addressBook>
  <owner>
    <cname>John Punin</cname>
    <email>puninj@cs.rpi.edu</email>
  </owner>
  <person>
    <cname>Harrison Ford</cname>
    <email>hford@famous.org</email>
  </person>
  <person>
    <cname>Julia Roberts</cname>
    <email>jr@pw.com</email>
  </person>
</addressBook>
```

## DTD:

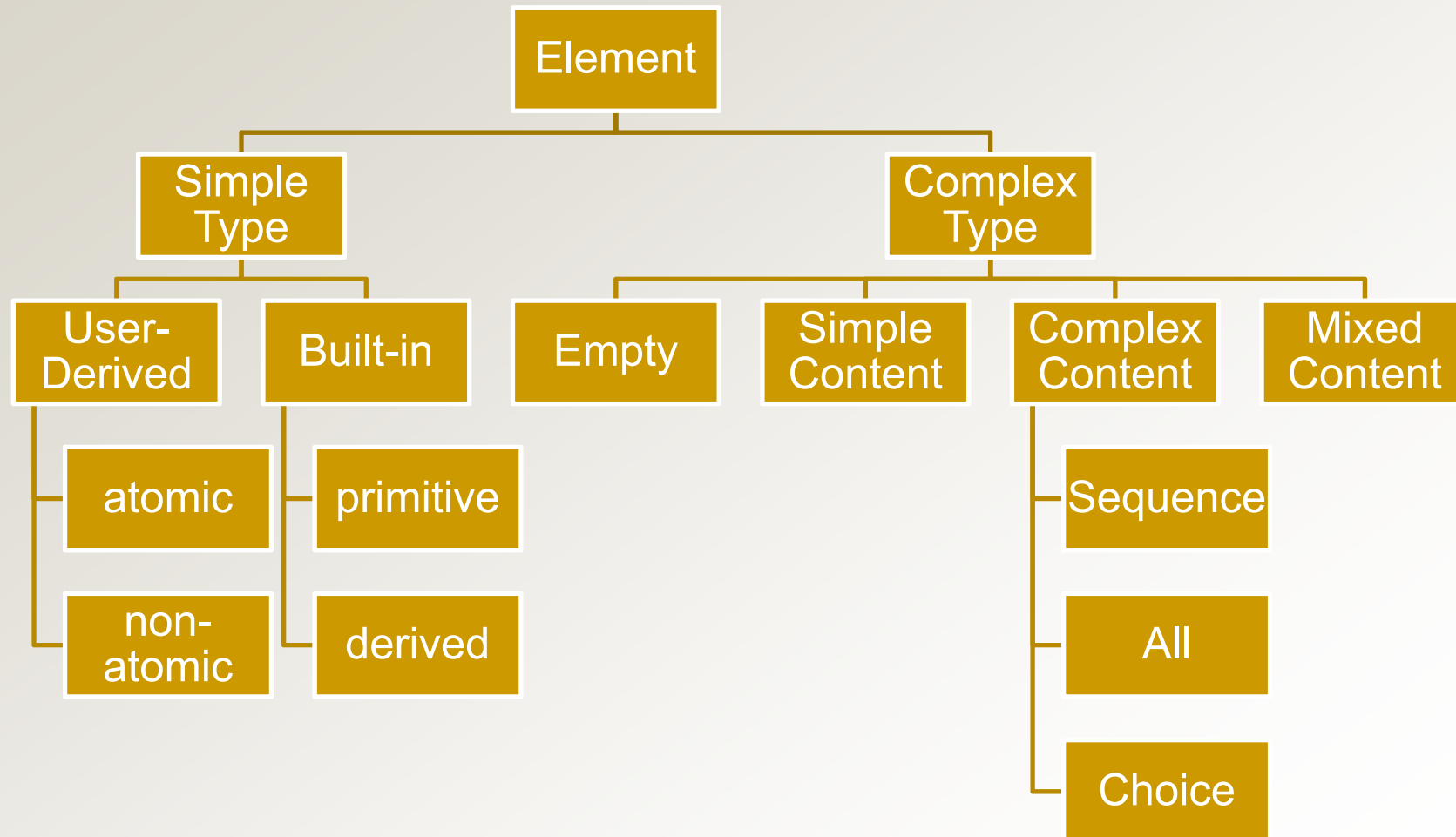
```
<!ENTITY % record (cname, email)>
<!ELEMENT addressBook (owner,person*)>
<!ELEMENT owner %record;>
<!ELEMENT person %record;>
```

## XML Schema:

```
<?xml version="1.0"?>
<xsd:schema
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:complexType name="record">
    <xsd:sequence>
      <xsd:element name="cname" type="xsd:string"/>
      <xsd:element name="email" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:element name="addressBook">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="owner" type="record"/>
        <xsd:element name="person" type="record"
          minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

# [ XML schema elemek ]



# [ Elemek megadása ]

```
<xs:element name="cim" type="xs:string"/>
```

# [Egyszerű típusok]

1. string
2. boolean
3. decimal
4. float
5. double
6. duration
7. dateTime
8. time
9. date
10. gYearMonth
11. gYear
12. gMonthDay
13. gDay
14. gMonth
15. hexBinary
16. base64Binary
17. anyURI
18. QName
19. NOTATION

# [Egyszerű XSD típusok – I.]

- `xs:string` - karakterlánc
- `xs:boolean` - true, false, 1, 0
- `xs:decimal` - tizedesjegyet is tartalmazó szám. Pl.: -5  
-3.7 2.88
- `xs:int` - egész szám, előjeles 32 bites
- `xs:integer` - egész szám (végtelen nagy lehet)
- `xs:positiveInteger` - pozitív egész szám (0 nem)
- `xs:negativeInteger` - negatív egész szám (0 nem)
- `xs:date` - Dátum. Formátuma: ÉÉÉÉ-HH-NN
- `xs:time` - Idő. Formátuma: óó-pp:mm.mm
- `xs:dateTime` - Dátum és idő.  
Formátuma: ÉÉÉÉ-HH-NN<sup>T</sup>óó-pp:mm.mm  
2007-03-05T08:26:00.00



# [Egyszerű XSD típusok – II.]

- `xsd:gMonth` – Naptári hónap (HH)
- `xsd:gYear` – Naptári év (ÉÉÉÉ)
- `xsd:gDay` – Naptári nap (NN)
- `xsd:gYearMonth` – Naptári év és hónap (ÉÉÉÉ-HH)
- `xsd:anyURI` – Hivatkozás. Pl.: <http://ade.web.elte.hu>
- `xsd:language` – Nyelv megadás. Pl. en-GB, fr, hu

# [ Megszorított típusok ]

- 1.normalizedString
- 2.token
- 3.language
- 4.NMTOKEN
- 5.NMTOKENS
- 6.Name
- 7.NCName
- 8.ID
- 9.IDREF
- 10.IDREFS
- 11.ENTITY
- 12.ENTITIES
- 13.integer
- 14.nonPositiveInteger
- 15.negativeInteger
- 16.long
- 17.int
- 18.short
- 19.byte
- 20.nonNegativeInteger
- 21.unsignedLong
- 22.unsignedInt
- 23.unsignedShort
- 24.unsignedByte
- 25.positiveInteger

# [ Simple Type megszorítások ]

- length - hossz
- minLength - min. hossz
- maxLength - max. hossz
- pattern - mintaillesztés reguláris kifejezéssel
- enumeration - felsorolás
- minInclusive -  $\geq$
- minExclusive -  $>$
- maxInclusive -  $\leq$
- maxExclusive -  $<$
- totalDigits - számjegyek száma
- fractionDigits - tizedesek száma

# [ Element paraméterek ]

- minOccurs
- maxOccurs
- default
- fixed
- id
- ref

# [ Elem előfordulás ]

- `minOccurs` – minimális előfordulás
- `maxOccurs` – maximális előfordulás
- Alapértelmezett értékük: 1.
- ```
<xsd:element name="ar" type="xs:integer" minOccurs="0" maxOccurs="unbounded"/>
```

unbounded: akárhányszor, akár el is hagyható.

# [ Alapértelmezés, kötelezőség ]

- **Alapértelmezett érték:**

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

- **Fix érték:**

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

- **Kötelező megadás:**

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

- **Opcionális megadás:**

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

# [ Saját típusok - megszorítások – I. ]

Értékhalmoz megadás:

```
<xs:element name="mennyi">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="10000"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

# [ Típusmegadás ]

xs:schema elem megadásakor kell definiálni.

```
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleType name="ArTipus">
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="10000"/>
    </xs:restriction>
  </xs:simpleType>
  <!-- további séma elemek ... -->
</xs:schema>
```

## Deklarálása

```
<xs:element name="AR" type="ArTipus"/>
```



# [ Megszorítások – II. ]

## ■ Felsorolás:

```
<xs:element name="szerzo">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="XY"/>
      <xs:enumeration value="ZV"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

# Megszorítások – III. - mintaillesztés

```
<xs:element name="tartalom">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="([a-z])*"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

- Kisbetűket, akármilyen hosszan.

# Mintaillesztés – reguláris kifejezések használata

- Legalább 1 kis- vagy nagybetű:

```
<xs:pattern value="([a-z][A-Z])+"/>
```

```
<xs:pattern value="([a-zA-Z])+"/>
```

- Felsorolás:

```
<xs:pattern value="Alma|Körte"/>
```

- Betű, szám 8 hosszan:

```
<xs:pattern value="[a-zA-Z0-9]{8}"/>
```

- ```
<xs:pattern value="\d{1}-\d{4}-\d{4}-\d{1}"/>
```

```
<ISBN>0-7891-4343-1</ISBN>
```

- \d - decimális karakter
- \s - nem megjelenő karakter: space, enter, tab...
- \D – nem decimális karakter

# [ Reguláris kifejezés - példák ]

- $a^*x$   $x, ax, aax, aaax \dots$
- $a?x$   $ax, x$
- $a+x$   $ax, aax, aaax \dots$
- $(a|b)+x$   $ax, bx, aax, abx, bax, bbx, aaax, aabx, abax, abbx, baax, babx$
- $[abcde]x$   $ax, bx, cx, dx, ex$
- $[a-e]x$   $ax, bx, cx, dx, ex$
- $^[0-9]x$  Nem számjegy, majd egy  $x$
- $.x$  Bármilyen karakter, majd egy  $x$
- $.*abc.*$   $1x2abc, abc1x2, z3456abchoor$

# [ Méret megszorítás ]

- **Fix hossz megadás:**

```
<xs:element name="jelszo">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:length value="8"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

- **Szám esetén: totalDigits**

- **Minimális és maximális hossz:**

```
<xs:minLength value="5"/>
<xs:maxLength value="8"/>
```

# [ Komplex típusok - I ]

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="konyv">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="szerzo"
          type="xs:string"/>
        <xs:element name="cim"
          type="xs:string"/>
        <xs:element name="ar"
          type="xs:integer"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

**xs:sequence → kötött sorrend**

# [ Komplex típusok - II ]

```
<xs:element name="konyv" type="konyvtipus" />
```

```
<xs:complexType name="konyvtipus" >
```

```
  <xs:sequence>
```

```
    <xs:element name="szerzo" type="xs:string"/>
```

```
    <xs:element name="cim" type="xs:string"/>
```

```
    <xs:element name="ar" type="xs:integer"/>
```

```
  </xs:sequence>
```

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

# [ Komplex típusok – III. ]

```
<xs:element name="employee" type="fullpersoninfo"/>
<xs:complexType name="personinfo">
  <xs:sequence>
    <xs:element name="firstname" type="xs:string"/>
    <xs:element name="lastname" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="fullpersoninfo">
  <xs:complexContent>
    <xs:extension base="personinfo">
      <xs:sequence>
        <xs:element name="address" type="xs:string"/>
        <xs:element name="city" type="xs:string"/>
        <xs:element name="country" type="xs:string"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```



# [Egyszerű tartalom]

```
<xs:element name="cipomeret">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:integer">
        <xs:attribute name="ország" type="xs:string"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

# [ Vegyes tartalmú elemek ]

- Szöveges tartalommal és gyermek elemmel egyaránt rendelkező elem deklarációja:

```
<xs:element name="CIM">  
  <xs:complexType mixed="true">  
    <xs:sequence>  
      <xs:element name="ALCIM"/>  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>
```

- Ennek megfelelő elem:

```
<CIM>Prison break<ALCIM>A szökés csak a  
  kezdet</ALCIM>  
</CIM>
```

# [ Komplex típusok – IV. ]

- `xs:choice` - Az alábbi FILM elem tartalmazhat egy RENDEZO vagy egy SZEREPLŐ elemet

```
<xs:element name="FILM">
```

```
  <xs:complexType>
```

```
    <xs:choice>
```

```
      <xs:element name="RENDEZO" type="xs:string"/>
```

```
      <xs:element name="SZEREPLŐ" type="xs:string"/>
```

```
    </xs:choice>
```

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

```
</xs:element>
```

- `xs:all` – tetszőleges sorrendben szerepelhetnek a gyermek elemek, de mindnek kell legalább egyszer
- `xs:sequence` és `xs:choice` elemek egymásba ágyazhatóak.
- Ezekben is használható a `minOccurs` és a `maxOccurs` jellemző.

# Attribútumok – jellemzők:

`xs:attribute`

- ~ elem megadáshoz

```
<xs:attribute name="alcim"  
type="xs:string" use="optional"/>
```

- Jellemzők:

- name
- type
- default
- fixed
- use: - optional, required, prohibited (tiltott)  
optional: alapértelmezett. default jellemző esetén  
ezt kell használni vagy a use-t elhagyni.

# [ Példa – 1: Tartalom és attribútum ]

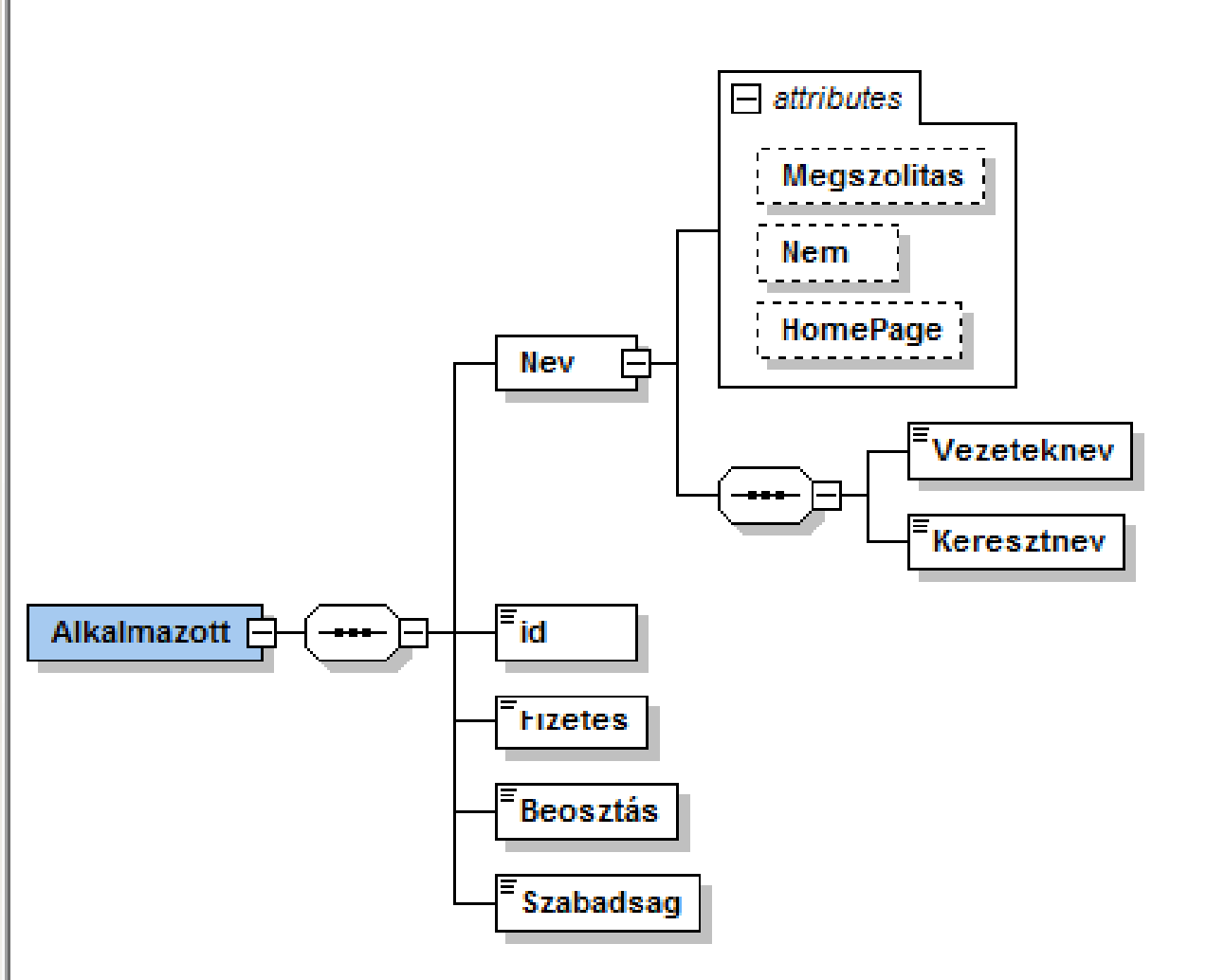
```
<xs:element name="cim">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="alcim" type="xs:string"
use="optional"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

# [ Szöveg és gyermek elemek ]

```
<xs:element name="level">
  <xs:complexType mixed="true">
    <xs:sequence>
      <xs:element name="nev" type="xs:string"/>
      <xs:element name="rendeles_azon" type="xs:positiveInteger"/>
      <xs:element name="szallitas" type="xs:date"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
<level>Tisztelt<nev>Citad Ella</nev>.
Az Ön <rendeles>1032</rendeles> számú rendelését várhatóan
<szallitas>2001-07-13</szallitas>-n szállítjuk.
</level>
```

# XSD készítés - 1.



# [ XSD készítés - 2. ]

```
<?xml version="1.0" encoding="ISO-8859-2"?>
<!DOCTYPE LELTAR SYSTEM "C:\###\ELTE\XML\4-Adatkotes\#alap\LeltarDTD.dtd">
<LELTAR xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="LeltarSema.xsd">
  <KONYV>
    <CIM>Fűszálak</CIM>
    <SZERZO>Walt Whitman</SZERZO>
    <KOTES>keménykötés</KOTES>
    <OLDALSZAM>462</OLDALSZAM>
    <AR>775 Ft</AR>
  </KONYV>
  <KONYV>
    <CIM>Az Álmos völgy legendája</CIM>
    <SZERZO>Washington Irving</SZERZO>
    <KOTES>olcsó kiadás papírkötés</KOTES>
    <OLDALSZAM>98</OLDALSZAM>
    <AR>295 Ft</AR>
  </KONYV>
```