

# BLM3580

# Sistem Programlama

---

2021-2022 GÜZ YARIYILI

DR.ÖĞR.ÜYESİ GÖKSEL BIRICIK

# XML

---

# eXtensible Markup Language: XML

---

Elektronik belge alışverişindeki problemleri çözmek için ortak bir standart olarak geliştirilmiştir.

1996'da W3C çalışma grubu geliştirdi.

Extensible Markup Language (XML) 1.0 (Second Edition), <http://www.w3c.org/TR/2000/REC-xml-20001006> , page 4

HTML gibi verinin nasıl gösterileceğinden ziyade, verinin nasıl tanımlanacağı ile ilgilenen bir dildir.

# XML Tasarım Hedefleri

---

İnternet üzerinden kullanılabilir olmalıdır.

Geniş bir uygulama yelpazesinde destek bulmalıdır.

SGML ile uyumlu olmalıdır.

XML belgelerini işleyen programların yazılması mümkün olduğunca kolay olmalıdır.

XML'e özel olan eklentiler mümkün olduğunca az, tercihen hiç olmamalıdır.

XML belgeleri insanlar tarafından kolay okunabilir nitelikte olmalıdır.

XML tasarımı hızla gerçekleştirilebilmelidir.

XML tasarımı belli kurallara bağlı (formal) ve kısa, özlü (concise) olmalıdır.

XML belgelerini yaratmak kolay olmalıdır.

XML komutlarının kısa ve öz olmasının önemi son derece düşüktür.

# eXtensible Markup Language: XML

---

HTML gibi verinin nasıl gösterileceğinden ziyade, verinin nasıl tanımlanacağı ile ilgilenen bir dildir.

HTML gibi sınırlı bir komut kümesine sahip değildir.

- Kullanıcı kendi ihtiyacına göre etiket tanımlayabilir.

XML, bilgileri bir metin dosyası olarak ifade etmek ile birlikte bunları bir yapı dâhilinde düzenlemektedir.

Veriyi tanımlamak ve verinin bu tanıma uygunluğunu kontrol edebilmek üzere DTD (Document Type Definition) kullanılır.

XML gerek duyulduğu yerlerde HTML kodları arasında da yer alabilir. Bu durum **Data Island** (veri adası) olarak adlandırılır.

# XML Elemanlarının Özellikleri

---

XML'de açılan her yönlendirme imi kapatılmalıdır.

XML büyük/küçük harf duyarlıdır.

XML elemanlarını içeren yönlendirme imleri açıldıkları sıranın tersinde kapatılmalıdır

XML elemanlarında mutlaka bütün yapının bağlı olduğu bir kök eleman olmalıdır

XML de kullanılan her türlü attribute çift tırnak içinde ifade edilmelidir.

XML de boşluklar (HTML ve Javascript'ten farklı olarak) korunmaktadır. Bu nedenle boşluklara dikkat edilmelidir.

XML de satır sonlarında yer alan CR/LF, LF'e dönüştürülmektedir.

# XML Rezerve Karakterler

---

Kullanılmak İstenen Karakter	Kullanılması Gereken Eşlenik
&	&amp;
<	&lt;
>	&gt;
'	&apos;
“	&quote;

# CDATA bloğu

XML parser'in değerlendirmesini istemediğimiz kısımlar için CDATA (character data) kullanılır.

CDATA bloğu içinde &, <, >, ' , " gibi XML için özel mana ifade eden karakterler kullanılabilir.

- XML içinde C kaynak kodu ilettiğinizi düşünün !
- CDATA bloğu ]]> ile sonlanır. Blok içinde yer almamalıdır.

Yollanmak istenen bilgi	XML kodlaması	CDATA kullanımı
if ( (a>b) && (b>c)) printf("a büyük sayı");	if ( (a &gt; b) &amp;&amp; (b &gt; c)) printf("&quote; a büyük sayı &quote;);	<![CDATA[ if ( (a>b) && (b>c)) printf("a büyük sayı"); ]]>



# XML Elemanlarının İsimlendirilmesi

---

İsimler harf ile başlamalıdır. Özel durumlarda \_ (alt çizgi) ve : (iki nokta üst üste) ile de başlayabilir.

İsimler ilk harfi takiben, harf, numara ve diğer karakterler ( \_ : ) ile devam edebilir.

İsimler xml, XML, Xml gibi harfler ile başlamamalıdır.

- (('X' | 'x') ( 'M' | 'm') ( 'L' | 'l')) harfleri standart yapı içinde ve XML'in ilerideki versiyonlarında kullanılabileceği düşünülerek kullanıcıların bu harflerin herhangi bir şekilde peş peşe gelebileceği isim vermemeleri istenmemektedir.

İsimler içinde boşluk bulunmamalıdır.

İsimlerin içinde “-” ve “.” sembollerinin kullanımından kaçınılmalıdır.

# Eleman ve Özellik Kullanımı

Öznitelikli Eleman tanımı	Özniteliğin Eleman ile ifade edildiği tanım
<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;sarkici cinsiyet="kadin"&gt;   &lt;ad&gt;Sezen&lt;/ad&gt;   &lt;soyad&gt;Aksu&lt;/soyad&gt; &lt;/sarkici&gt;</pre>	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;sarkici&gt;   &lt;cinsiyet&gt;kadin&lt;/cinsiyet&gt;   &lt;ad&gt;Sezen&lt;/ad&gt;   &lt;soyad&gt;Aksu&lt;/soyad&gt; &lt;/sarkici&gt;</pre>

`<araba modeli="2016"/>`

# Özellik Kullanımı

---

Genel olarak özellik kullanımından mümkün olduğunca kaçınmak gereklidir.

Öznitelikler birden fazla değer içeremezler. Ancak aynı işlem alt-elemanlar kullanılarak yapılacak olursa bu kısıtlama ortadan kalkar.

Özniteliklerin sayısını arttırmak bunları değerlendirmek durumunda olan programların işini güçleştirmektedir.

Özniteliklerin DTD'ye uygunluklarının kontrol edilmesinde zorluklar vardır.

Öznitelikler yapısal bir ifade şekli değildir.

# XML Namespace

---

Belirli bir komut kümesi yok. Farklı bilgi bloklarını herkes aynı isimle ifade edebilir. İsim toplulukları ile karışıklık organize edilebilir.

```
<?xml version="1.0"?>
```

```
<kitaplık xmlns:kitap = "http://www.ce.yildiz.edu.tr/xmlns-kitap"
```

```
xmlns:periyodik = "http://www.ce.yildiz.edu.tr/xmlns-periyodik">
```

```
<kitap:cilt ciltno= "1">
```

```
<kitap:konu>Algoritmalar</kitap:konu>
```

```
</kitap:cilt>
```

```
<periyodik:cilt ciltno= "Aug-2021/2">
```

```
<periyodik:konu>Computing Survey</periyodik:konu>
```

```
<periyodik:yayinci>ACM Press</periyodik:yayinci>
```

```
</periyodik:cilt>
```

```
</kitaplık>
```

# XML Namespace

---

İsim toplulukları, kök elemanın öznitelikleri olarak tanımlanmıştır.

- `http://www.ce.yildiz.edu.tr/xmlns-kitap` ve `http://www.ce.yildiz.edu.tr/xmlns-periyodik` değerleri URI (Uniform Resource Identifier) olarak adlandırılmaktadır.
- İsim topluluklarına neden URI tanımlarız? menşei belirtmek ve benzerinin oluşmasını engellemek için.
- İsim topluluklarına verilen değerler (URI) hiçbir zaman XML ayrıştırıcısı (parser) tarafından ziyaret edilerek bu adresten bilgi almak amacıyla kullanılmaz. Bunlar sadece isim topluluğunun bir diğerinden ayrılmasını sağlayacak birer metindir.

Bazı özel durumlarda aksi söylenmedikçe isimlerin varsayılan isim topluluğu (default namespace) kullanması arzu edilebilir

```
<kitaplık          xmlns:          = "http://www.ce.yildiz.edu.tr/xmlns-kitap"
                  xmlns:periyodik = "http://www.ce.yildiz.edu.tr/xmlns-periyodik">
```

# Doğru düzenlenmiş (well formed) ve geçerli (valid) XML belgeleri

---

Doğru formda olmak: XML için geçerli yazım kurallarına uygun olarak yazılmış olmak.

Geçerli olmak: XML belgesinin bağlı bulunduğu DTD veya Schema'ya bağlı olarak değerlendirildiğinde arzu edilen düzenleme kurallarına uyması.

- Geçerli belge her zaman doğru formdadır. Tersi doğru olmayabilir.

XML ile işlem yapabilmek için XML ayrıştırıcısı gerekir. Bunlar geçerleyen ve geçerlemeyen şeklinde olabilir. Geçerleyenler:

Ayrıştırıcı Adı	Açıklama
MSXML	Microsoft XML parser
XERCES	Apache XML's project parser
JAXP	Microsystem's Java API XML Parsing
XML4J	IBM's XML for Java
Oracle XML parser	Oracle XML parser
XMLBooster	
SXP	The Silfide XML Parser

# XML-DTD

---

# XML DTD (Document Type Definition)

---

Yazım kurallarına uygun olsa bile XML belgeleri kurallara uygun yapıda düzenlenmelidir.

Geçerleyen XML ayrıştırıcıları DTD'yi okuyup XML'in buradaki tanımlara uygun olup olmadığını denetler.

- MS XML Validator, XMLwriter, XMLSpy, ...

DTD, XML'in başında (internal) ya da harici bir dosyada (external) olabilir.

## XML Dosyası içinde (Internal)

```
<!DOCTYPE root_element [  
DTD_Eleman_tanımlamaları ]>
```

## XML Dosyası dışında (External)

```
<!DOCTYPE root_element  
SYSTEM "DTD_dosya_ismi">
```



# DTD Eleman Tanımları

---

`<!ELEMENT eleman_ismi ( eleman_içeriği ) >`

*eleman\_içeriği* için genel olarak #PCDATA (Parsed Character Data) veya başka bir alt-eleman tanımı kullanılabilir.

- İçinde özel amaçlı karakterler (< , > , & gibi) varsa ayrıştırıcı bunları yorumlar. Gerekli dönüşümlerin yapılmış olması önemlidir.

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

`<!ELEMENT kitap (yazar, yayıncı) >`

`<!ELEMENT yazar (#PCDATA) >`

`<!ELEMENT yayıncı (#PCDATA) >`

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

`<kitap>`

`<yazar>Ali & Ayşe</yazar>`

`<yayıncı>ABC yayınevi</yayıncı>`

`</kitap>`

# DTD Eleman Tanımları

---

EMPTY, boş bir eleman tanımını olup kendi içinde hiçbir karakter bulundurmadığı gibi alt-eleman da bulundurmaz.

- `<!ELEMENT hr EMPTY> → <hr/>`

ANY, #PCDATA tanımına uygun her türlü metnin yanı sıra, tanımlı olan elemanlardan da oluşan herhangi bir dizilişteki içeriği barındırabilir.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT yazar (#PCDATA) >
<!ELEMENT detay ANY>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<yazar>Ali & amp; Ayşe </yazar>
<detay>
Sayın<yazar>Ali</yazar> & amp;
<yazar>Ayşe</yazar> tarafından kaleme alınmış bu
eser...
</detay>
```

# DTD Eleman Sayı Tanımları

DTD Tanımlaması	Açıklama
<!ELEMENT sınıf (öğretmen, öğrenci) >	<i>sınıf</i> isimli eleman altında bir tane <i>öğretmen</i> ve bir tane de <i>öğrenci</i> isimli eleman olabilir.
<!ELEMENT sınıf (öğretmen, öğrenci+) >	<i>sınıf</i> isimli eleman altında bir tane <i>öğretmen</i> ve bir ya da daha fazla <i>öğrenci</i> isimli eleman olabilir.
<!ELEMENT tatlı (baklava   dondurma) >	<i>tatlı</i> isimli eleman altında bir tane <i>baklava</i> veya bir tane <i>dondurma</i> isimli eleman olabilir.
<!ELEMENT yemek (çorba?, meze*, anayemek, meşrubat+) >	<i>yemek</i> isimli eleman altında <i>çorba</i> olmayabilir, olursa bir tane olur. <i>meze</i> olmayabilir, olursa kaç tane olacağı belli değildir. <i>anayemek</i> mutlaka olacaktır. <i>meşrubat</i> ise en az bir tane olacaktır.
<!ELEMENT albüm (ismi, (şarkı, süresi)+)>	<i>albüm</i> isimli eleman altında bir tane <i>ismi</i> elemanı ve en az bir tane olmak üzere ( <i>şarkı</i> , <i>süre</i> ) ikilisinden oluşan elemanlar olacaktır.
<!ELEMENT fırın EMPTY>	<i>fırın</i> isimli eleman boş olarak tanımlanmıştır. < <i>fırın</i> /> şeklinde

# DTD Eleman Sayı Tanımları

DTD Tanım satırı	Tanıma uygun XML gösterimi
<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;!ELEMENT mesajlar (#PCDATA   mesaj)*&gt; &lt;!ELEMENT mesaj (#PCDATA)&gt;</pre>	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;mesajlar&gt;Çarşamba gününün mesajları     &lt;mesaj&gt;Acele evi arayın&lt;/mesaj&gt;     &lt;mesaj&gt;Bankaya uğrayınız &lt;/mesaj&gt; &lt;/mesajlar&gt;</pre>
<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;!ELEMENT çiftlik (çiftçi+,(köpek*   kedi?), keçi*, (tavuk+   ördek*)&gt; &lt;!ELEMENT çiftçi (#PCDATA)&gt; &lt;!ELEMENT köpek (#PCDATA)&gt; &lt;!ELEMENT kedi (#PCDATA)&gt; &lt;!ELEMENT keçi (#PCDATA)&gt; &lt;!ELEMENT tavuk (#PCDATA)&gt; &lt;!ELEMENT ördek (#PCDATA)&gt;</pre>	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;çiftlik&gt;     &lt;çiftçi&gt;Tahir&lt;/çiftçi&gt;     &lt;çiftçi&gt;Ayşe&lt;/çiftçi&gt;     &lt;kedi&gt;Tekir&lt;/kedi&gt;     &lt;tavuk&gt;Benekli&lt;/tavuk&gt; &lt;/çiftlik&gt;</pre>

# 1 DTD, 2 Geçerli XML

DTD Tanım satırı	Tanıma uygun XML gösterimi
<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;!ELEMENT ders     ( sınıf, (öğretmen   asistan+),öğrenci+)&gt; &lt;!ELEMENT sınıf (#PCDATA)&gt; &lt;!ELEMENT öğretmen (#PCDATA)&gt; &lt;!ELEMENT asistan (#PCDATA)&gt; &lt;!ELEMENT öğrenci (#PCDATA)&gt;</pre>	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;ders&gt;     &lt;sınıf&gt;D108&lt;/sınıf&gt;     &lt;öğretmen&gt;Rıfki Beşyıldız&lt;/öğretmen&gt;     &lt;öğrenci&gt;Ali Türk&lt;/öğrenci&gt;     &lt;öğrenci&gt;Ayşe Elibol&lt;/öğrenci&gt;     &lt;öğrenci&gt;Fatma Kara&lt;/öğrenci&gt; &lt;/ders&gt;</pre> <pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;ders&gt;     &lt;sınıf&gt;D108&lt;/sınıf&gt;     &lt;asistan&gt;Ahmet Varol&lt;/asistan&gt;     &lt;asistan&gt;Mahmut Gürbüz&lt;/asistan&gt;     &lt;öğrenci&gt;Ali Türk&lt;/öğrenci&gt;     &lt;öğrenci&gt;Ayşe Elibol&lt;/öğrenci&gt;     &lt;öğrenci&gt;Fatma Kara&lt;/öğrenci&gt; &lt;/ders&gt;</pre>

# DTD ile Özelliklerin Tanımlanması

---

Öznitelikler elemanların içinde yer alır.

- Bu nedenle bir özniteliğin tanımının olabilmesi için onu kullanan bir eleman tanımının bulunması gereklidir.

Öznitelikler temel olarak 3 ayrı tipte tanımlanabilir.

- En yaygını CDATA (karakter dizisi) olarak değer alanlardır.
- Ayrıca TOKENIZED ve ENUMERATED olarak iki farklı tipi de vardır.
- İçinde &, <, >, ' , " gibi özel karakterler yer almamalıdır.

Birden fazla özniteliği olan elemanlarda özniteliklerin yazılış sırasının herhangi bir önemi yoktur.

```
<!ATTLIST eleman_ismi öznitelik_ismi CDATA #REQUIRED>
```

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

```
<!ELEMENT kişi EMPTY>
```

```
<!ATTLIST kişi cinsiyet CDATA #REQUIRED)>
```

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

```
<kişi cinsiyet = "erkek" />
```

# DTD ile Özniteliklerin Kontrolü

DTD Tanım satırı	Açıklama
<!ATTLLIST <i>adres postakodu</i> CDATA # <b>REQUIRED</b> >	<i>adres</i> isimli elemanın mutlaka <i>postakodu</i> özniteliğinin değeri verilmiş olmalıdır.
<!ATTLLIST <i>adres postakodu</i> CDATA # <b>IMPLIED</b> >	<i>adres</i> isimli elemanın <i>postakodu</i> özniteliğinin değeri verilmeyebilir.
<!ATTLLIST <i>adres postakodu</i> CDATA # <b>FIXED</b> "34220">	<i>adres</i> isimli elemanın <i>postakodu</i> özniteliği tanımlı olmasa bile değeri mutlaka 34220 olacak, bunun dışındaki değerler belgenin geçerli olmasını engelleyecektir.

# “Tokenized” Özellikler

---

XML yazarına özelliklerin değerlerini sınırlandırma imkanı sağlar.

ID bir elemanın değerinin bir şekilde belirlenmesini sağlamak için kullanılır.

IDREF ise ID ile belirlenen elemana referans verilmesini sağlamak amacıyla kullanılır.

- XML belgesinin doğruluğunu kontrol eden bir ayrıştırıcı (parser) her ID için XML belgesinden bir IDREF karşılığı olup olmadığını kontrol edecektir.

Bir XML belgesi içinde birden fazla ID kullanımı mümkün değildir.

- Tanımlanmamış bir ID'ye referans veren IDREF kullanımı hata yaratır.

ID için #FIXED de kullanılamaz.



# Tokenized Özellikler – ID ve IDREF

---

*book* elemanının *shippedBy* isimli özniteliği, *shipping* elemanının *shipID* özniteliği için kullanılan değerler olan *s1* ve *s2* değerlerini alabilirler. Bunların dışında verilecek herhangi bir değer ID olarak tanımlanmadığı için IDREF olarak kullanılması mümkün değildir. Bu durum oluşursa ayrıştırıcı hata verir.

```
<?xml version= "1.0"?>
<!DOCTYPE bookstore [
    <!ELEMENT bookstore (shipping+, book+)>      <!ELEMENT shipping (duration)>
    <!ATTLIST shipping shipID ID #REQUIRED>      <!ELEMENT book (#PCDATA)>
    <!ATTLIST book shippedBy IDREF #IMPLIED>      <!ELEMENT duration (#PCDATA)>
]>
<bookstore>
    <shipping shipID = "s1"><duration>2 to 4 days</duration></shipping>
    <shipping shipID = "s2"><duration>1 day</duration></shipping>
    <book shippedBy = "s2">Java How to Program </book>
    <book shippedBy = "s1">How to Program </book>
    <book shippedBy = "s2">XML How to Program </book>
</bookstore>
```

# Tokenized Özellikler - ENTITY

---

ENTITY ile tanımlanan değerler ayrıştırıcı tarafından belge içerisine kullanılabilir.

```
<?xml version= "1.0"?>
```

```
<!DOCTYPE mail [
```

```
    <!ELEMENT mail (from, to*,subject?,body)>
```

```
    <!ELEMENT from (#PCDATA)>
```

```
    <!ELEMENT to (#PCDATA)>
```

```
    <!ELEMENT subject (#PCDATA)>
```

```
    <!ELEMENT body (#PCDATA)>
```

```
    <!ENTITY replyto "gbiricik@yildiz.edu.tr">
```

```
<mail>
```

```
    <from>&replyto;</from> → çıktıda <from>gbiricik@yildiz.edu.tr</from>
```

```
    <to>sabri@yildiz.edu.tr</to>
```

```
    <to>mustafa@yildiz.edu.tr</to>
```

```
    <subject>test mesajı</subject>
```

```
    <body>
```

```
        Bu mesaj deneme amaçlı yollanmıştır. Lütfen siliniz. </body>
```

```
</mail>
```

# Tokenized Özellikler - NMTOKEN

---

NMTOKEN (name token), içeriğinde harfler, sayılar, nokta, altçizgi, ara çizgi ve iki nokta üst üste gibi karakterlerin yer aldığı sınırlandırılmış bir yapıyı belirler.

Özellikle telefon numarası, posta kodu gibi bilgilerin belirlenen forma uygun olarak girilmesini sağlamak üzere kullanılır.

<!ATTLIST *birim telefon* NMTOKEN #REQUIRED>

<*birim telefon* = "212-383-5730"> Bilgisayar Mühendisliği Bölümü </*birim*>

# Enumeration Özellik Tipleri

DTD Tanım satırı	Açıklama
<code>&lt;!ATTLIST kişi cinsiyet ( E   K ) "E" &gt;</code>	kişi isimli elemanın cinsiyet olarak belirlenen attribute'ü E veya K değerlerinden birini alabilir. Önceden belirlenmiş değer olarak ise E verilmiştir.
<code>&lt;!ATTLIST kişi cinsiyet ( E   K ) #IMPLIED&gt;</code>	cinsiyet özniteliği için önceden belirlenmiş bir değer yoktur. cinsiyet özniteliği belli olamayan türde elemanlar için buna benzer bir tanımlama yaparak XML belgesini değerlendirecek uygulamaya esneklik sağlanmıştır.
<code>&lt;!ATTLIST kitap referans NOTATION ( Java   C ) "C"&gt;</code>  <code>&lt;!NOTATION C SYSTEM "http://www.prenticehall.com/books/C"&gt;</code>	kitap isimli elemanın referans olarak tanımlı özniteliği Java veya C olarak tanımlı değerlerden birini alabilmekle beraber ilk değer olarak C verilmiştir. NOTATION kullanılarak C ile ilişkilendirilmiş bir tanım var olduğu için referans isimli özniteliğin değeri <a href="http://www.prenticehall.com/books/C">http://www.prenticehall.com/books/C</a> olarak kabul edilir.

# XML-DTD Örneği

---

```
<?xml version="1.0"?>
<!-- Dosya: ogrenci.xml Aşağıdaki xml dosyası ogrenci.dtd isimli dosyada tanımlı DTD özelliklerine göre değerlendirilecektir -->

<!DOCTYPE Ogrenciler SYSTEM "ogrenci.dtd">
<Ogrenciler>
    <Ogrenci>
        <Adi>Ali</Adi>
        <Soyadi>Muhterem</Soyadi>
        <Bolum Kodu="011">Bilgisayar</Bolum>
        <Adres Sehir="Ist">Acibadem</Adres>
        <Telefon Tipi="Ev">216-1112233</Telefon>
        <Telefon Tipi="Cep">666-4445566</Telefon>
    </Ogrenci>
    <Ogrenci>
        <Adi>Mahmut</Adi>
        <Soyadi>Perisan</Soyadi>
        <Bolum Kodu="012">Elektrik</Bolum>
        <Adres Sehir="Ist">Besiktas</Adres>
        <Telefon Tipi="Ev">212-1112233</Telefon>
    </Ogrenci>
</Ogrenciler>
```

# XML-DTD Örneği

---

```
<?xml version="1.0"?>
```

```
<!-- Dosya: ogrenci.dtd -->
```

```
<!-- ogrenci-dtd.xml dosyasının DTD uygunluğunu kontrol eder -->
```

```
<!ELEMENT Ogrenciler (Ogrenci)*>
```

```
<!ELEMENT Ogrenci (Adi,Soyadi,Bolum,Adres,Telefon*)>
```

```
<!ELEMENT Adi ( #PCDATA )>
```

```
<!ELEMENT Soyadi ( #PCDATA )>
```

```
<!ELEMENT Bolum ( #PCDATA )>
```

```
<!ATTLIST Bolum Kodu CDATA #REQUIRED>
```

```
<!ELEMENT Adres ( #PCDATA )>
```

```
<!ATTLIST Adres Sehir CDATA #FIXED "Ist">
```

```
<!ELEMENT Telefon ( #PCDATA )>
```

```
<!ATTLIST Telefon Tipi CDATA #REQUIRED>
```