



XML-Technologien

Tutorium 3

Themen:

Musterlösung Ü2
XML-Schema

von:

Kain Kordian Gontarska
kainkordian@gmail.com

XML-Parser

- Lesen das Dokument ein und stellen der darüber liegenden Anwendung die gewonnenen Informationen zur Verfügung.
- Werden nach 2 Kriterien unterschieden:
 - validierender / nicht validierender Parser
 - Angebotene Schnittstelle (SAX, DOM) zum Zugriff auf das XML-Dokument
 - SAX:
 - Behandelt das zu parsende Dokument als Anreihung von Events. Nach Abschluss eines Events, sind die Informationen nicht mehr erreichbar.
 - Fehler werden in der Tiefe erkannt.
 - DOM:
 - Erstellt beim Parsen einen DOM-Baum auf dem nach dem Parsen zugegriffen werden kann. Der DOM-Baum kann wieder in einem XML-Dokument gespeichert werden.
 - Fehler werden in der Breite erkannt.

XML - Schema

- Gibt uns die Möglichkeit spezielle Typen zu definieren (Für die Elemente, Attribute, Attributwerte etc.)
- Es kann von bestehenden Typen geerbt werden (Typvererbung, NICHT OOP – Vererbung!). Dabei erweitern oder einschränken eines bestehenden Typs
- SimpleType und ComplexType
 - SimpleType:
 - Atomare Datentypen, Listen und Unions (Keine Attribute oder Kindelemente)
 - ComplexType:
 - Bietet die Möglichkeit Elementstrukturen zusammenhängend zu definieren. Kindelemente und Attribute können hier definiert werden.

XML – Schema SimpleType

- Enthalten:
xs:string, xs:decimal, xs:integer, xs:float, xs:boolean, xs:date, xs:time
- XML-spezifisch:
Qname, anyURI, language, ID, IDREF

```
<xs:simpleType name="monatInt">  
  <xs:restriction base="xs:integer">  
    <xs:minInclusive value="1"/>  
    <xs:maxInclusive value="12"/>  
  </xs:restriction>  
</xs:simpleType>  
<xs:simpleType name="monate">  
  <xs:list itemType="monatInt"/>  
</xs:simpleType>
```

```
<monate>  
  1 2 3 4 5 6 7 8 9 10 11 12  
</monate>
```

XML – Schema SimpleType

- Einem SimpleType kann eine Auswahl an weiteren SimpleTypes gegeben werden.

```

<xs:simpleType name="monatsname">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Jan"/>
    <xs:enumeration value="Feb"/>
    <xs:enumeration value="Mär"/>
    <!-- und so weiter ... -->
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="monat">
  <xs:union memberTypes="monatsname
monatInt"/>
</xs:simpleType>

```

```

<monat>Jan</monat>
<monat>4</monat>

```

XML – Schema ComplexType

- Ergänzend zu den einfachen Typen bieten komplexe Typen die Möglichkeit den Datentypen explizit Elemente und Attribute anzuhängen.

```
<xs:complexType name="pc-Typ">
  <xs:sequence>
    <xs:element name="name" type="xs:string"/>
    <xs:element name="hersteller" type="xs:string"/>
    <xs:element name="prozessor" type="xs:string"/>
    <xs:element name="mhz" type="xs:integer" minOccurs="0"/>
    <xs:element name="kommentar" type="xs:string" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:integer"/>
</xs:complexType>
```

Kindelemente können auf 3 unterschiedliche Arten kombiniert werden:

xs: sequence, xs: choice, xs:all

XML – Elemente

- Mithilfe der Datentypen lassen sich nun Einschränkungen über die Definition der Elemente beschreiben.

```
<xs:element name="pc-liste">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="pc" type="pc-Typ" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
<pc-liste>
  <pc>
    <name>Dimension 3100 </name>
    <hersteller>Dell</hersteller>
    <prozessor>AMD</prozessor>
    <mhz>3060</mhz>

    <kommentar>Arbeitsplatzrechner</kommentar>
  </pc>
</pc-liste>
```

XML – Schema

- XML-Schemata sind wohlgeformte XML-Dokumente und beginnen mit einer Deklaration.

```
<?xml version="1.0"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
        xmlns:bsp="http://de.wikipedia.org/wiki/XML_Schema#Beispiel"
        targetNamespace="http://de.wikipedia.org/wiki/XML_Schema#Beispiel">
<!--.....-->
</schema>
```

- Referenzierung eines XML-Schemas.

```
<root
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  Xsi:noNamespaceschemaLocation="myXSD.xsd">
</root>
```