INFORMATION

(1) HTML
Der Einsatzbereich und die Eigenschaften der Hypertext Markup Language (HTML) sind bekannt

(2) XML
Die eXtensible Markup Language (XML) kann zur Erstellung von gültigen (validen) XML-Beschreibungen eingesetzt werden

(3) XML_JAVA
Die Schnittstelle Java API for XML Processing (JAXP) kann verwendet werden, um XML-Beschreibungen in Java-Programmen angemessen behandeln zu können

HyperText Markup Language (HTML)

(1) HTML ist das vom W3C standardisierte und im World Wide Web (WWW) verwendete Dokumentformat

(2) Eigenschaften
  (1) Eine Anwendung der Standard Generalized Markup Language (SGML)
  (2) Strukturierung des Textes durch Markierungen
  (3) Verknüpfung innerhalb und zwischen Dokumenten durch Hypertext-Verweise
  (4) Einbindung von Graphiken

(3) Anwendungsmöglichkeiten
  (1) Einfache Textdokumente mit Abbildungen
  (2) Vernetzte Hypertext-Dokumente
  (3) Hypertext-Mail
  (4) Maschinenunabhängige Schnittstelle zu Informationssystemen

Aufbau eines HTML-Dokuments

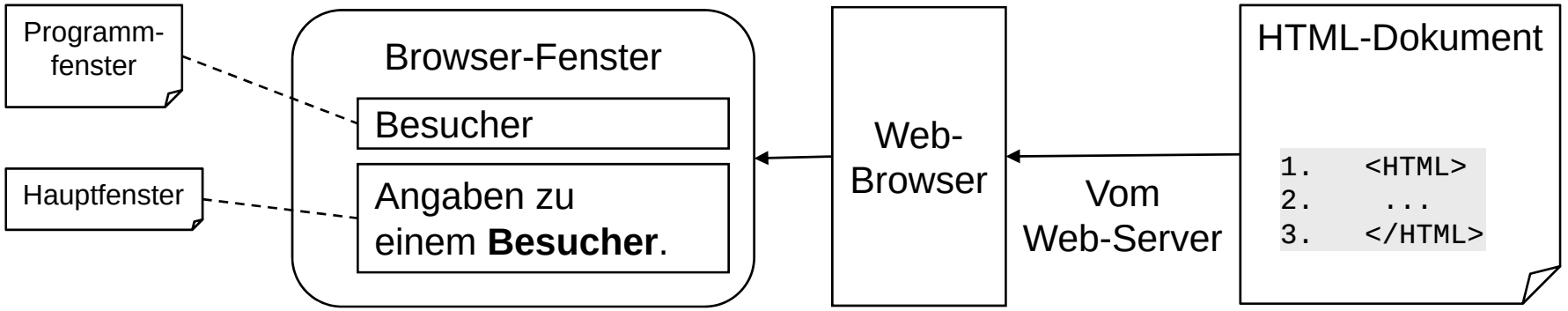
```
1.   <!-- besucher.html -->
2.   <html>                              <!-- Start-Tag des HTML-Elements "html" -->
3.    <head>                             <!-- Start-Tag des HTML-Elements "head" -->
4.     <title>Besucher</title>           <!-- HTML-Element "title" -->
5.    </head>
6.    <body>
7.     <p>Angaben zu einem <b>Besucher</b>.</p>
8.    </body>
9.   </html>                             <!-- Ende-Tag des HTML-Elements "html" -->
```

(1) HTML-Elemente
   (1) Besteht aus Start-Tag, Elementinhalt und Ende-Tag

(2) Aufbau eines HTML-Dokuments
   (1) Ein HTML-Element <html>, das aus den HTML-Elementen <head> und <body> besteht

(3) Weitere im Beispiel verwendete Elemente
   (1) <!-- … --> Kommentar
   (2) <title>…</title> Dokumententitel
   (3) … Paragraph (Absatz)
   (4) … Bold (Fettdruck)

Darstellung eines HTML-Dokuments im Browser



(1) Der Web-Browser setzt die im HTML-Dokument auftretenden Elemente um

(2) Klassifizierung der Typen von HTML-Elementen
   (1) Dokumentenstruktur
      (1) Linearer Text
      (2) Hypertext
   (2) Schriftauszeichnung
      (1) Physisch (typographisch)
      (2) Logisch (idiomatisch)

Die wichtigsten HTML-Tags im Überblick

<html> ...</html>	Deklaration der in HTML beschriebenen Web-Seite
<head> ... </head>	Abgrenzung des Seitenkopfes
<title> ... </title>	Festlegung des Titels
<hn> ... </hn>	Abgrenzung einer Überschrift auf der n. Ebene
...	Setzen von ... in Fettdruck
...	Setzen von ... in Kursivdruck
...	Klammerung einer ungeordneten Spiegelliste
...	Klammerung einer nummerierten Liste
<menu> ... </menu>	Klammerung eines Menüs von -Einträgen
	Start eines Listeneintrags
 	Zeilenumbruch
<p>	Beginn eines Paragraphen
<img src="...">	Einfügen eines Bildes
 ... 	Festlegung eines Hyperlink

Hypertext-Verknüpfung

(1) Die wirkliche Stärke von HTML ist die Verknüpfung von Dokumenten über das Internet hinweg

(2) Die Hypertext-Verknüpfung ist das Schlüsselkonzept des Web

(3) Die Verknüpfung erfolgt durch die Ergänzung von Ankern im HTML-Dokument

(4) Es besteht die Möglichkeit, auf Anker in beliebigen HTML-Dokumenten zu verweisen

    (1) Es lassen sich intuitive Informationsflüsse erzeugen

HTML-Dokument mit Liste und Hypertext

(1) Mittels eines Hypertext-Ankers (<a>) lässt sich eine bestimmte Stelle festlegen, auf die vom selben oder einem anderen Hypertext-Dokument verwiesen werden kann

(2) Start (href) bzw. Ziel (name) werden im Anker-Element durch ein Attribut ausgedrückt

   (1) name="Name des Zielankers" (Definition eines Ankers)

   (2) href="URL#Name des Zielankers" (Verweis auf einen Anker)

```
1.   <!-- besucherliste.html -->
2.   <html>
3.    <head><title>Besucherliste</title></head>
4.    <body>
5.     <ul>
6.      <li>Thompson</li>
7.      <li><a name="Anker1">Miller</a></li>
8.      <li>Reynold</li>
9.      <li>Hier folgen ggf. weitere Namen von Besuchern</li>
10.     <li>Cramer hat einen Bezug zu <a href="#Anker1">Miller</a></li>
11.    </ul>
12.   </body>
13.  </html>
```

LZ HTML – ÜA EIGENSCHAFTEN

(1) HTML ist eine Spezialisierung von
   (1) UML
   (2) SGML
   (3) XML

(2) HTML ist zur Beschreibung von beliebigen Datenstrukturen gut geeignet
   (1) Ja, weil _____
   (2) Nein, weil _____

(3) Was kann als eine besonders charakteristische Eigenschaft von HTML angesehen werden?
   (1) Beschreibung des Layouts von Dokumenten
   (2) Verwendung von Tags
   (3) Verknüpfung zwischen Dokumenten
   (4) Sprachumfang

HTML-Formulare

(1) Sind Bestandteil des HTML-Standards zur Übermittlung von Daten an den Web-Server

(2) Eigenschaften
   (1) Definiert durch ein "form"-Element
      (1) Besitzt die Attribute "action" und "method"
   (2) Es stehen die folgenden Formular-Elemente zur Verfügung
      (1) "input": Vielseitigstes Eingabeelement
      (2) "textarea": Verschiebbares Eingabefenster
      (3) "select": Auswahlliste
      (4) "option": Element innerhalb der "select"-Auswahlliste

eXtensible Markup Language (XML)

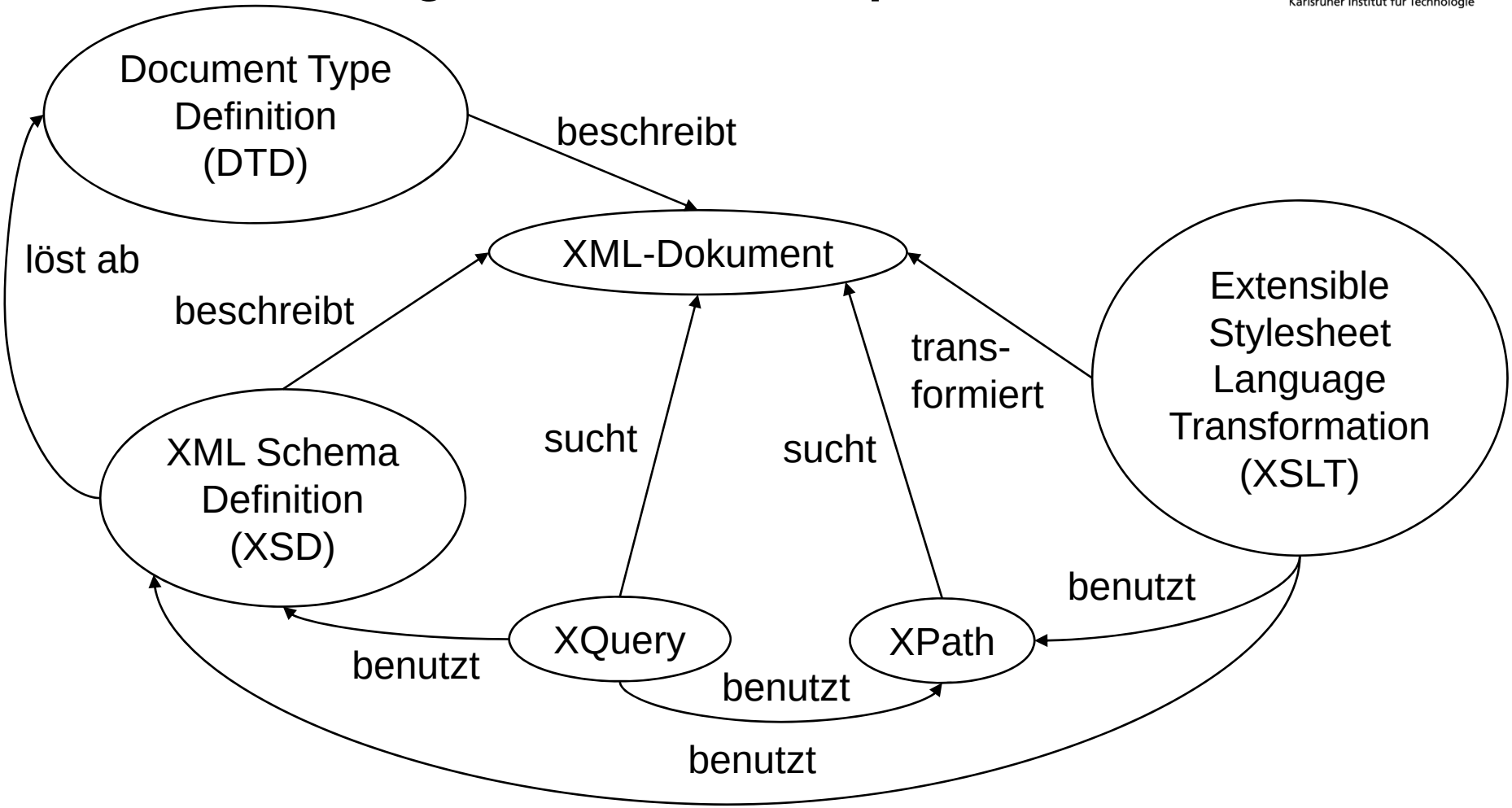
(1) Die eXtensible Markup Language XML ist ein W3C-Standard, durch den beliebige Datentypen beschrieben werden können
   (1) XML kann als eine Metasprache angesehen werden

(2) Zwei besondere Eigenschaften
   (1) Konzept des Dokumententyps
   (2) Portabilität

(3) Wichtigste Unterschiede zwischen HTML und XML
   (1) XML beschreibt nicht die Darstellung der Daten
   (2) In XML haben die Daten eine Bedeutung
   (3) XML wird von Programmen zur gemeinsamen Nutzung und zum Austausch von Daten genutzt
   (4) XML ist erweiterbar

(4) Eine genaue Kenntnis von XML ist eine Voraussetzung für das Verständnis von Web-Services

Zusammenhang zwischen XML-Spezifikationen



(1) Der Fokus liegt auf Spezifikationen, die für die (serviceorientierte) Anwendungsintegration relevant sind

Struktur eines XML-Dokuments

(1) Ein XML-Dokument beginnt mit einem sog. Prolog
  (1) Dokumentendeklaration
    (1) Beispiel

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

  (2) Verarbeitungsbefehle

(2) Dokumenteninstanz folgt auf den Prolog
  (1) Wurzelelement des Dokuments
  (2) Kindelemente
    (1) Kann weitere Kindelemente enthalten
    (2) Kann einen Wert haben oder nicht
    (3) Kann Attribute enthalten
  (3) Attribute
    (1) Eigenschaften, durch die zusätzliche Information zu einem Element geliefert wird

XML-Namen und Konventionen

(1) Die Namensvergabe betrifft insbesondere die Bezeichnung von Elementen und Attributen

(2) Syntaktische Vorgaben
   (1) Der Aufbau wird durch EBNF-Regeln vorgegeben
   (2) Das erste Zeichen darf keine Zahl sein
   (3) Viele Sonderzeichen dürfen in Namen nicht verwendet werden

(3) Es wird empfohlen, in Namen "bedeutungsvolle" Wörter zu verwenden und auf "symbolische" Zeichen zu verzichten

(4) Konventionen zur konsistenten Benennung von Elementen und Attributen
   (1) Das erste Zeichen ist ein Kleinbuchstabe
   (2) Worttrennung in einem Element- oder Attributnamen erfolgt durch die Verwendung eines Großbuchstabens

Beispiel eines XML-Dokuments

```
1.  <?xml version= "1.0" encoding = "UTF8" ?>
2.  <!– Ein einfaches XML-Dokument -->
3.
4.  <visitor id="v4711">
5.   <firstName>John</firstName>
6.   <lastName>Doe</lastName>
7.   <address>
8.    <street>Zirkel 2<street>
9.    <city>Karlsruhe</city>
10.   <postalCode>76128</postalCode>
11.  </address>
12. </visitor>
```

(1) Der Prolog besteht aus den ersten zwei Zeilen

(2) Das Wurzelelement ist <visitor>
   (1) Zugewiesenes Attribut "id"
   (2) Kindelemente <name>, <yearLastVisit>, <address>

(3) Das XML-Dokument ist konform zu den XML-Regeln
   (1) Es wird als wohlgeformt bezeichnet

LZ XML – ÜA STRUKTUR

(1) Warum kann XML als eine Metasprache angesehen werden?

(2) In welchen Zeilen treten im XML-Beispiel-dokument die folgenden Bestandteile auf

- (1) Erstes Kindelement
- (2) Erstes Attribut
- (3) Prolog
- (4) Epilog
- (5) Wurzelelement
- (6) Dokumentinstanz
- (7) Dokumentdeklaration

```
1.  <?xml version= "1.0" encoding = "UTF8" ?>
2.  <!– Ein einfaches XML-Dokument -->
3.
4.  <visitor id="v4711">
5.   <firstName>John</firstName>
6.   <lastName>Doe</lastName>
7.   <address>
8.    <street>Zirkel 2<street>
9.    <city>Karlsruhe</city>
10.   <postalCode>76128</postalCode>
11.  </address>
12. </visitor>
```

Namensräume

(1) XML-Dokumente sollen mehrere Auszeichnungsvokabulare nutzen können
   (1) Idee der Wiederverwendung von einem Vokabular durch mehrere diese verarbeitende Software-Module

(2) Vokabulare sollen von verschiedenen Seiten definiert werden können
   (1) Keine zentrale Koordination
   (2) XML-Namensräume liefern eine einfache Möglichkeit zur Qualifizierung von Elementtypen und Attributnamen

(3) Deklaration eines Namensraums
   (1) Durch das reservierte Attribut "xmlns = <someURI>" wird der sog. Default Namespace deklariert
   (2) Durch "xmlns:<prefix> = <someURI>" wird der Bezeichner <prefix> für den URI eingeführt
      (1) Der qualifizierte Name "<prefix>:<element name>" drückt aus, dass der Elementname zu diesem Namensraum gehört

Alternative Namensraum-Festlegungen zu dem XML-Beispieldokument

```
1.  <?xml version= "1.0" encoding = "UTF8" ?>
2.  <visitor id="v4711"
3.   xmlns="http://www.WASA.edu/visitorInfo">
4.   <firstName>John</firstName>
5.   <lastName>Doe</lastName>
6.   <address xmlns=http://www.WASA.edu/addr>
7.     <street>Zirkel 2<street>
8.     <city>Karlsruhe</city>
9.     <postalCode>76128</postalCode>
10.   </address>
11. </visitor>
```

```
1.  <?xml version= "1.0" encoding = "UTF8" ?>
2.  <vi:visitor id="v4711"
3.   xmlns:vi="http://www.WASA.edu/visitorInfo"
4.   xmlns:ad="http://www.WASA.edu/address">
5.   <vi:firstName>John</vi:firstName>
6.   <vi:lastName>Doe</vi:lastName>
7.   <ad:address>
8.    <ad:street>Zirkel 2<ad:street>
9.    <ad:city>Karlsruhe</ad:city>
10.   <ad:postalCode>76128</ad:postalCode>
11.  </ad:address>
12. </vi:visitor>
```

(1) Default Namespace
  (1) Unübersichtlich bei verschränkten Elementen, die zu unterschiedlichen Namensräumen gehören

(2) Qualifizierte Namen (QNames)
  (1) Bestehen aus dem durch ein Präfix angegebenen Namensraum und den lokalen XML-Namen

LZ XML –
ÜA NAMENSRÄUME

(1) Die Angabe eines Namensraums erfolgt im XML-Dokument als
   (1) Teil des Prologs
   (2) ein Element
   (3) ein Attribut
   (4) ein Attribut des Wurzelelements

(2) Qualifizierte Namen (QNames)
   (1) setzen sich zusammen aus _____
   (2) treten im Zusammenhang mit den "Default Namespaces" auf
   (3) werden genutzt, um ein XML-Dokument übersichtlicher zu gestalten

Cooperation & Management (C&M, Prof. Abeck)
Institut für Telematik, Fakultät für Informatik

Motivation zur Festlegung der Struktur

(1) Ziel: Angemessene Einschränkung des Freiheitsgrads der auszutauschenden XML-Nachrichten

(2) Beispiel: Unterschiedliche Möglichkeiten der Beschreibung der gleichen Information

/1/
```
1. <schulung>Web – Einführung in
2.           Web-Technologien</schulung>
```

/2/
```
1. <schulung>
2.  <titel>Einführung in Web-Technologien</titel>
3.  <kurztitel>Web</kurztitel>
4. </schulung>
```

/3/
```
1. <schulung titel="Einführung in Web-Technologien"
2.           kurztitel="Web">
3. </schulung>
```

Beschreibung der Struktur eines XML-Dokuments

(1) Alternative Ansätze
   (1) Document Type Definition (DTD)
   (2) XML Schema Description (XSD)

(2) Ein XML-Dokument, das die (durch DTD oder XSD vorgegebenen) Strukturvorgaben erfüllt, wird als gültig oder valide (engl. valid) bezeichnet

(3) DTD war der erste vom W3C verfolgte Ansatz
   (1) Basiert auf der Backus-Naur-Form
   (2) Beispiel

```
1.  <!DOCTYPE visitor [
2.   <!ELEMENT visitor (firstName, lastName, address)>
3.   <!ELEMENT firstname (#PCDATA)>
4.   <!....>
5.  ]>
```

XML Schema Description (XSD)

(1) XSD weist einige wichtige Vorteile gegenüber der DTD auf
   (1) Verwendet selbst XML
   (2) Höhere Mächtigkeit
   (3) Unterstützung von Datentypen und Namensräumen

(2) Durch XSD beschreibbare Aspekte
   (1) Elemente und Attribute, die im XML-Dokument auftreten
   (2) Element-Hierarchie einschließlich der Anzahl und Reihenfolge von Kindelementen
   (3) Datentypen und Gültigkeitsbereich von Werten der Elemente und Attribute

Beispiel einer XSD

```
1.  <?xml version="1.0"?>
2.  <xs:schema
3.   xmlns:xs="http://www.w3.org/2001/XMLSchema"
4.   xmlns:vi="http://www.WASA.edu/visitorInfo"
5.   targetNamespace="http://www.WASA.edu/visitorInfo">
6.   <xs:import namespace="http://www.WASA.edu/addr" schemaLocation="addrType.xsd" />
7.
8.   <xs:element name="visitor" type ="visitorType" />
9.   <xs:complexType name="visitorType">
10.    <xs:sequence>
11.     <xs:element name="firstName" type="xs:string"/>
12.     <xs:element name="lastName" type="xs:string"/>
13.     <xs:element name="address" type="addr" /> <!- Imported type from addrType.xsd -->
14.    </xs:sequence>
15.    <xs:attribute name ="id" type="xs:string" />
16.   </xs:complexType>
17.  </xs:element>
18. </xs:schema>
```

(1) Ein XSD-Dokument ist selbst ein XML-Dokument

- (1) "xmlns:xs=..." ist der Namensraum zur Beschreibung von XSDs
- (2) "targetNamespace" legt den Namensraum für die im vorliegenden XSD-Dokument definierten Elemente fest

LZ XML –
ÜA STRUKTURBESCHREIBUNG

(1) Wie heißen die zwei wichtigsten Ansätze zur Beschreibung der Struktur eines XML-Dokuments?

(2) Was ist der entscheidende Unterschied zwischen den beiden Ansätzen?

(3) Handelt es sich bei den beiden Ansätzen um Sprachen und wenn ja, um welche Art von Sprachen?

(4) Welche der folgenden XML-Elemente sind Teil der XSD und was drücken diese Elemente aus?

   (1) element
   (2) DOCTYPE
   (3) sequence
   (4) record
   (5) complexType

XML und Java

(1) Extrahieren, Erzeugen und Validieren von XML-Dokumenten mittels Java

(2) Zwei alternative APIs stehen zur Verfügung
   (1) Simple API for XML (SAX)
   (2) Document Object Model (DOM)

(3) "Document" ist die zentrale Klasse bei DOM
   (1) Erzeugung einer Instanz dieser Klasse über Verwendung der Klassen "DocumentBuilderFactory" und "DocumentBuilder"
   (2) Lesen des Inhalts mittels der Klasse "XPath"
      (1) Stellt u.a. eine Methode "evaluate(String XPathExpression, Document doc)" zur Verfügung
   (3) Verwendung der durch die Java API for XML Processing (JAXP) bereitgestellten Interfaces zum Aufbau bzw. zur Veränderung des Inhalts eines "Document"-Objekts

Java XML API (JAXP)

(1) Die wichtigsten sieben JAXP-Interfaces und deren Beziehungen