

Aufgabe 1a)

[4 Punkte]

Erzeugen Sie ein detailliertes¹ XML Schema für folgendes XML Dokument:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <node id="1" name="root-Knoten">
    <node ref="2" role="must-branch"/>
    <node ref="3" role="optional"/>
  </node>
  <node id="2" name="left_child">
    <node ref="4" type="optional"/>
  </node>
  <node id="3"/>
  <node id="4"/>
</root>
```

Passen Sie das o.g. XML Dokument ggf. an, damit dies auf ihr erstelltes Schema referenziert!

Aufgabe 1b)

[2+2 Punkte]

Erweitern Sie ihr XML Schema aus Aufgabe 1a), indem Sie für **id** einen Schlüssel, für **ref** ein Constraint definieren.

Aufgabe 1c)

[2 Punkte]

Die **role** muss innerhalb des Element-Kontextes eindeutig sein!

¹ Detailliert bedeutet hier, dass das Schema die Definitionen der beteiligten Elemente beinhalten muss. Eine Definition nach Folie 29 des Foliensatzes ist nicht ausreichend detailliert:

```
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="root"/>
</xs:schema>
```

Laden Sie die Datei `class.xml` von der Institutsseite oder dem Moodle herunter. Sie modelliert einen Java-Quelltext (dessen Verständnis für die Bearbeitung dieser Aufgabe jedoch nicht von belang ist).

Eine sehr gute Dokumentation zu XPath (und später: XQuery)-Funktionen finden Sie unter:

<http://www.w3.org/TR/xpath-functions/>

Allgemein wichtig sind hier die Kapitel 7 (String-Manipulationen) und 15 (insb. Abschnitt 15.4). Hinweis: in dem Dokument werden Sie Funktionen mit Namensraum-Präfix (`fn: . . .`) finden. Wenn Sie die Funktionen, z. B. Mit Oxygen, testen, lassen Sie diese bitte einfach weg!

Funktionen mit dem Präfix `op:` können Sie **nicht** direkt aufrufen (siehe den letzten Absatz des Abschnitts „Namespaces and Prefixes“: <http://www.w3.org/TR/xpath-functions/#namespace-prefixes>). Diese Funktionen werden auch nicht für die Lösung der Aufgaben benötigt!

Schreiben Sie XPath Ausdrücke:

Beispielaufgabe 2)

[0 Punkte]

Selektieren sie in Großbuchstaben alle internen Ids für Klassen.

Lösung:

```
self::node()/descendant-or-  
element(class)/attribute(internalID)/upper-  
case(.)
```

oder in Kurzform:

```
//class/@internalID/upper-case(.)
```

Warum?

`//class` → „für alle Elemente mit dem Namen class“ ...

`@internalID` → verweise auf das Attribut internalID ...

`/upper-case(.)` → und gebe den aktuellen Kontext (der Punkt, verweist in diesem Fall auf das Attribut internalID) in Großbuchstaben zurück.

Aufgabe 2a)

[1 Punkt]

Selektieren sie alle Pakete (Attribut `package`), die benutzt werden.

Aufgabe 2b) [2 Punkte]

Geben Sie alle Interfaces in dem Paket `java.util` zurück!

Aufgabe 2c) [2 Punkte]

Geben Sie Methoden zurück, die mit „get“ anfangen.

Aufgabe 2d) [2 Punkte]

Geben Sie alle Namen der Methoden mitsamt ihrem primitiven Rückgabetypen in der Form:

`<Methodenname>():<primitive_type>`

zurück. (Methoden, die Klassen/Objekte zurückliefern, können ausgespart werden).

Aufgabe 2e) [3 Punkte]

Gegeben sei die XML-Dokumentenstruktur auf Vorlesungsfolie 55/65. Ist es möglich, die Abfrage umzudrehen und ausgehend von selektierten Autoren die Namen der Bücher zurückzugeben? Wenn ja: schreiben sie den XPath-Ausdruck. Wenn nicht: Warum?

Allgemeiner Bearbeitungshinweis:

Geben Sie zu ihren Lösungen stets auch eine grobe Lösungsskizze mit an, damit ihre Ergebnisse nachvollziehbar sind!

Geben Sie die Lösung bis zum 21.06.2011, 8:15 Uhr ab, indem Sie sie entweder in Fach 45 einwerfen (links dreimalig als Buch getackert!), oder bevorzugt per E-Mail mit dem Betreff-prefix `[is2]` an `busche@ismll.de` schicken. Sie erhalten eine Eingangsbestätigung bei E-Mailabgabe.

Bonusaufgabe 1)

[10 Punkte]

Erstellen Sie ein XML-Dokument, dessen XPath-Ausgaben folgende Werte annehmen:

count(//file)	3
count(//directory)	3
count(/directory)	0
count(/root)	1
count(node()/directory)	2
count(//@*)	14
count(//*)	10
count(//linkS)	2
count(//linkC)	1
count(//@name)	7
string(/)	“
//@name	{„root“, „verzeichnis1“, „file1“, „dir1“, „verzeichnis2“, „file1“, „file2“}
boolean(//directory/@name)	true
boolean(//linkS/@name)	false
node-name(//file/parent::node())	<dieser XPath gibt einen Fehler!>
node-name(//linkC/parent::node())	„root“

Bonusaufgabe 2)

[3+2+2 Punkte]

Gegeben sei die Datei `class.xml` der Aufgabe 2. Lösen Sie folgende Fragen mit einem (1) XPath Ausdruck:

- Geben Sie alle Klassen und Interfaces zurück, die min. 3 Methoden besitzen/deklarieren.
- Geben Sie die Namen aller Klassen an, die eine Methode besitzen, welche ihrerseits einen int als Übergabeparameter haben.
- Geben Sie alle Klassen zurück, die statisch false zurückgeben.