2. Versuch: XSLT

Aufgabe 2.1 (XML to HTML with XSLT)
Lösen Sie Aufgabe 1.4 mit XSLT.

Aufgabe 2.2 (Recursion in Data)

- Schreiben Sie ein XSLT-Skript, das die Struktur der Meere und Flusssysteme aus Mondial abbildet: Der Elementinhalt jedes sea-Elementes soll dessen Namen, sowie alle Flüsse, die in das Meer fliessen, enthalten. Weiterhin soll jedes solche river-Element den Namen des Flusses, seine Länge, sowie alle seine direkten Nebenflüsse enthalten. Analog, falls der Fluss einen See entwässert.
  Beispiel:

```
<waters>
  <sea>
    <name>North Sea</name>
    <river>
      <name>Rhein</name>
      <length>...</length>
      <river>
        <name>Main</name>
        <length>...</length>
        <river>
          <name>Tauber</name>
          <length>...</length>
        </river>
        :
      </river>
      <river>
        <name>Neckar</name>
        <length>...</length>
        <river>
      </river>
      :
    </river>
  </sea>
</waters>
```

(b) Schreiben Sie ausserdem ein zweites Stylesheet, das aus dem obigen File für jeden Fluss, der in ein Meer fließt, die Summe der Längen aller seiner (direkten oder transitiven) Nebenflüsse berechnet und in einer Tabelle ausgibt.

(c) Schreiben Sie ein weiteres Stylesheet (welches mondial.xml als Eingabe verwendet), das für jeden Fluss die Summe der Länge aller Flüsse, die (direkt oder indirekt) in ihn fließen, als Tabelle ausgibt.

(d) Erweitern Sie das Stylesheet aus Teil (a) so, dass mit einer XPath-Anfrage an das Ergebnis die folgenden Anfragen beantwortet werden können:
  - Kann es sein, dass Regen südlich des Äquators fällt, und dann ins Mittelmeer fließt?
  - Von welchen Ländern fließt Wasser in das Schwarze Meer?

Aufgabe 2.3 (XSLT and Recursive Data Structures: Binary Search Trees)

a) Überlegen Sie sich ein XML-Markup zur Repräsentation von *Binären Suchbäumen* (verwenden Sie z.B. die Definition des entsprechenden Abstrakten Datentyps aus *Informatik I*). Geben Sie die Repräsentation des leeren BSB an, sowie eine nicht-leere Beispielinstanz.

b) Geben Sie die DTD dazu an. Geben Sie optional auch ein XML Schema an.

c) Schreiben Sie XSL-Stylesheets um (i) Werte in den Baum einzufügen, (ii) die in dem Baum gespeicherten Werte in Inorder auszugeben, (iii) den Baum grafisch (z.B. durch geschachtelte HTML-Tabellen) auszugeben, (iv) Suche nach einem Wert, und (iii) Löschen eines Wertes aus dem Baum (Aufruf z.B. durch `saxonXSL -o mybsb.xml mybsb.xml bsb-insert.xsl insert=6`).

Aufgabe 2.4 (Stylesheet for Schedules)

Schreiben sie ein XSL-Stylesheet das Termindaten im unten angegebenen Schema in einen Terminkalender in Form einer HTML-Tabelle umwandelt:

```
<?xml version="1.0" encoding="UTF-8"?>
<schedule name="Meine Termine">
  <year n="2010">
    <month n="1">
      <day n="16">
        <entry starttime="T15:00" duration="PT02:00">
          <name>Institutsbesprechung</name>
          <description>monatliche Sitzung, Tagesordnung zz noch nicht bekannt</description>
          <location>Sitzungszimmer</location>
        </entry>
      </day>
    </month>
    <month n="5">
      <day n="2">
        <entry starttime="T18:00" duration="PT00:50">
          <name>Telefonat</name>
        </entry>
      </day>
    </month>
  </year>
</schedule>
```

Der Terminkalender sollte in etwa so aussehen:

jan	feb	mar	apr	may	jun	jul	aug	sep	oct	nov	dec
1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2 18:00 - Telefonat	2	2	2	2	2	2
3 15:00 - Institutsbesprechung	3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4	4	4
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮