



Praktikum zur Veranstaltung XML-Technologie: **Übung 12**

APIs für XML-Daten



Organisatorisches



- Arbeitsverzeichnis:
`~/lv/xmltech/12/`
- Dateinamen:
`08-bestell.xml, 08-bestell.dtd (unverändert)`
`10-tempest.xml, 10-shaksper.dtd (dito),`
`telbuch_streaming.{c|rb},`
`TelBuchStreaming.java, MeinHandler/*`
- Abzugeben:
`12-count.{c|rb}|Count12.java,`
`12-sort.{c|rb}|Sort12.java`
- Werkzeuge:
`(X)Emacs; gcc | javac, java | ruby; libxml2`



- Aufgabenbeschreibung (Übersicht)
 - Teil A: Anwendung eines *streaming*-API (SAX o.ä.)
 - Teil B: Anwendung eines DOM-artigen API
- Wahl der Programmiersprache
 - Für jeden Aufgabenteil haben Sie die Wahl zwischen den Programmiersprachen C, Java und Ruby
 - Für den SAX-Teil A finden Sie Beispielcode für alle drei Fälle im Ordner `~/lv/xmltech/12/` (`~ = /local0/werntges`)
 - Hinweise zur Kompilierung finden Sie als Kommentare in den Beispielen zu C und Java
 - Für den DOM-Teil B sind Sie auf die Vorlesungsbeispiele bzw. eigene Recherchen angewiesen!
- Hinweise
 - Geben Sie jeweils nur die Lösung für eine der drei Sprachen ab!
 - Mischen ist erlaubt, z.B. die Abgabe von `12-count.c` und `12-sort.rb`



Übung 12: A – SAX-artiges Arbeiten



- Teil A
 - Schreiben Sie ein Programm, das die Datei `10-tempest.xml` einliest und auf `stdout` eine Liste der Sprecher und mit der Anzahl ihrer Reden (Vorkommnisse von Element „SPEAKER“ mit diesem Namen in „SPEECH“) ausgibt.
 - Beispiel für eine der gewünschten Ausgabezeilen:
`ADRIAN sprach 9 mal`
 - Bedingung:
Einsatz von SAX bzw. analoger *Callback*-Technik
 - Abzugebende Datei:
 - C: `12-count.c`
 - Ruby: `12-count.rb`
 - Java: `Count12.java`



Übung 12: B – DOM-artiges Arbeiten



- Teil B
 - Schreiben Sie ein Programm, das die Datei `08-bestell.xml` einliest, als internes Dokument im Speicher vorhält und daraus ein umsortiertes Dokument erzeugt.
 - Geben Sie dieses Dokument auf `stdout` aus.
 - Die 5 enthaltenen Bestellungen sollen dabei in der Reihenfolge absteigender Bestellnummern erscheinen:
Zuerst Bestellung mit `id=„myIDstring04“`
 - Abzugebende Datei:
 - C: `12-sort.c`
 - Ruby: `12-sort.rb`
 - Java: `Sort12.java`
 - Bedingung: Der Einsatz von XSLT ist hier nicht erlaubt!



Übung 12: Abschließende Bemerkungen



- Kommentare
 - Für „C“ wird die Verwendung von „libxml2“ empfohlen.
Dokumentation dazu unter: <http://www.xmlsoft.org/>
Hinweis: Die Behandlung des DOM-Teils mit „C“-Mitteln könnte komplizierter geraten als mit den anderen Programmiersprachen...
 - Für Java finden Sie Beispielcode in der Vorlesung, wobei zu DOM zumindest etliche Codefragmente vorliegen.
 - Die Lösungen in Ruby sind besonders kompakt (36 bzw. 12 Codezeilen). Beide basieren auf Klassen aus dem Modul REXML.
 - Beachten Sie die Demos zu Beginn der Übung
- Abgabe
 - Normal, zu Beginn der nächsten (und letzten) Übung.
- Hinweis zum Schluss
 - **Die letzte Übung ist reserviert für die Klärung Ihrer Fragen. Bringen Sie daher Fragen, Fallbeispiele etc. mit!**