

XML – Aufgaben zu Merkblatt 3

1 Grundbegriffe

Die folgenden Begriffe und gegebenenfalls das englischsprachige Pendant sollten Sie verstehen, einordnen und erklären können: anonymer Typ, atomarer Typ, einfacher Typ, global deklariertes Element, komplexer Typ, lokal deklariertes Element, XML-Schema.

2 XML-Schema schreiben

Geben Sie zu folgendem XML-Dokument ein XML-Schema an. Beachten Sie die zusätzlichen Kommentare im XML-Dokument. Bei nicht näher angegebener Struktur haben Sie Freiheitsgrade. Das XML-Dokument muss in jedem Fall gültig hinsichtlich Ihres XML-Schemas sein.

```
<rechneruebung name="XML"> <!-- Attribute ist verpflichtend -->
  <aufgabe> <!-- beliebig viele aufgabe-Elemente-->
    <!-- mindestens ein student-Element
    <student matrikelnummer="60000"> <!-- Attribute optional -->
      <!-- genau ein name gefolgt von entweder einem bestanden oder
      nicht-bestanden-Element -->
      <name>Arnold Adams</name>
      <bestanden/>
    </student>
    <student matrikelnummer="60001">
      <name>Anna Albers</name>
      <bestanden/>
    </student>
  </aufgabe>
  <aufgabe>
    <student matrikelnummer="60000">
      <name>Arnold Adams</name>
      <nicht-bestanden/>
    </student>
  </aufgabe>
</rechneruebung>
```

3 Pattern

Einfache Typen können mit regulären Ausdrücken (pattern) weiter eingeschränkt werden. Es gibt die folgenden Operatoren mit den aus der Theoretischen Informatik bekannten Seman-

tik: | (entweder oder), + (ein oder mehr), * (kein, ein oder mehr), ? (optional). Ausdrücke können mit (und) geklammert werden. Zeichen, sofern Sie keine spezielle Bedeutung wie + haben, werden einfachen hingeschrieben. Für Zeichen mit spezieller Bedeutung müssen Fluchtsymbole verwendet werden: \+, \\, \?, \(.

Beispiel: (a+|(ba* c?)+). Beachte: das Leerzeichen gilt als Zeichen. Dieser Ausdruck matched zum Beispiel baa c, a, baa und baa cb .

Mit {n,m} lassen sich minimale untere Anzahl Vorkommen und obere angeben: a{7,11} (sieben bis elf as). Mit [12a-e\ -] lässt sich eine Menge von Zeichen angeben. Der Bindestrich ist ein nur hier verwendbarer Operator, der alle Zeichen von linke bis rechten Operand aufzählt. Der Ausdruck ist gleichbedeutend mit 1|2|a|b|c|d|e|\ -. ^ berechnet das Komplement einer Menge. Er darf nur vor [...] verwendet werden. ^[0-9] matched jedes Zeichen außer arabische Ziffern. Mit #xFF oder #x1F0 lassen sich beliebig Unicode Zeichen zur Basis 16 angeben.

Geben Sie jeweils ein Pattern an, der die im folgenden beschriebenen Zeichenketten matched.

1. Nicht leere Zeichenketten beliebiger Länge, die immer abwechselnd aus drei Zeichen bestehen: arabischer Ziffer 0-9, Kleinbuchstaben a-z und Großbuchstaben A-Z. Beispiel: 5hD, 1uZ5hD, 1uZ5hD9zZ.
2. Alle Zeichenketten, die weder Leerzeichen noch Fragezeichen enthalten.
3. Ganzzahlige Java-Literale in Oktalrepräsentation. Inklusive Unterstrichen: 0444_111, 07, 0_0.

4 Simple Type

Geben Sie einen W3C Schema-simpleType mit Hilfe eines regulären Ausdrucks an, der Deutsche KFZ-Kennzeichen umfasst. Dieses besteht für den zivilen Bereich aus zwei Teilen:

1. Ein bis drei Großbuchstaben (Unterscheidungszeichen) und
2. Ein bis zwei Großbuchstaben gefolgt von ein bis vier Ziffern (Erkennungsnummer). Bei einem Oldtimer ist ein H angehängt.

Ein KFZ-Kennzeichen umfasst maximal acht Zeichen inklusive dem H-Zeichen bei Oldtimern. Für diese Aufgabe soll zwischen Unterscheidungszeichen und Erkennungsnummer immer genau ein Leerzeichen sein. Zwischen den Buchstaben und der Zahl der Erkennungsnummer soll genau ein Bindestrich sein.

Einige Beispiele:

KA SL-1234

RA ST-123H

HSK TT-1

B Z-1