

Programmieren 3 – Grundlagen der Webprogrammierung

Zur Erlangung eines **Leistungsnachweises** müssen während des Semesters folgende Leistungen erbracht werden:

- a) Zu mindestens drei Übungsaufgaben von verschiedenen Aufgabenblättern während der Übungen eine korrekte Lösung vorstellen und auf Rückfragen zufriedenstellend antworten
- b) Lösen und präsentieren **aller** Assignments

Aufgabenblatt 1

1.1

- a) Bearbeiten Sie eine der folgenden zwei Varianten (wobei erste Variante empfohlen wird):
 - (i) Legen Sie unter der Entwicklungsumgebung [IntelliJ¹](#) für sich ein Projekt „XML *IhrName*“ an.
 - (ii) Legen Sie unter der Entwicklungsumgebung [Eclipse²](#) für sich ein Projekt „XML *IhrName*“ an.
Installieren Sie den XML-Editor [oXygen](#) (am besten als Plugin von Eclipse); die passende Lizenz für die **Version 25.x** finden Sie auf dem ELLI-Server.
- b) Erstellen Sie unter Ihrem Projekt eine XML-Datei mit folgendem Inhalt:

```
<?xml version="1.1" encoding="UTF-8"?>
<!DOCTYPE hello>
<hello>Hello folks!</hello>
```
- c) Erweitern Sie das vorige XML-Dokument um ein Element mit dem Namen `special` und dem Inhalt `<<<<< &&&&& >>>>>`. Welche unterschiedlichen Möglichkeiten gibt es, diesen Inhalt einzutragen?
Laden Sie XML-Dokumente mit unterschiedlichen Realisierungsvarianten des `special`-Elements in einen Web-Browser³. Gibt es bei der Darstellung im Browser Unterschiede zwischen den Realisierungsvarianten?
- d) Erweitern Sie das Element `hello` um ein beliebiges Attribut und fügen Sie in der Datei außerdem an einer Stelle Ihrer Wahl einen Kommentar hinzu.
- e) Verdeutlichen Sie sich anhand Ihres XML-Dokuments der vorigen Aufgaben die Begriffe „XML-Deklaration“, „Element“, „Attribut“, „Markup-Deklaration“.
- f) Erweitern Sie Ihr XML-Dokument um ein zweites Wurzelement `nochEinHello`. Wie sieht jetzt die Darstellung des Dokuments im Browser aus? Warum?

¹ Verwenden Sie am besten die „Ultimate Edition“ über eine kostenlose Lizenz für Studierende.

² Verwenden Sie am besten die Variante „Eclipse IDE for Enterprise Java and Web Developers“ in der Version „2023-09“.

³ Verwenden Sie am besten einen Browser der Mozilla-Familie (z. B. Firefox).

1.2

Laden Sie vom ELLI-Server die Datei `AndroidManifest.xml`, fügen Sie diese in Ihr XML-Projekt ein und bestimmen Sie deren Inhalt: Welche Elementtypen sind enthalten, wie viele Instanzen von jedem Typ kommen vor, gibt es leere Elemente, Container-Elemente, Text-Elemente, Elemente mit Mixed Content? Welche Attribute, Kommentare kommen vor, ...? .

1.3

Laden Sie vom ELLI-Server die Datei `fehlerhaft.xml` und fügen Sie diese ebenfalls in Ihr XML-Projekt ein. Beseitigen Sie alle dort enthaltenen Fehler.

Programmieren 3 – Grundlagen der Webprogrammierung

Aufgabenblatt 2

2.1

- a) Entwerfen Sie eine einfache, aber dennoch sinnvolle DTD für ein XML-Dokument, das folgenden Sachverhalt abbildet:

Ein Projekt besteht aus mehreren Mitarbeitern. Jeder Mitarbeiter hat einen Namen und evtl. einen Vornamen sowie eine eindeutige Personalkennung (Schlüssel). Außerdem hat jeder Mitarbeiter verschiedene Fähigkeiten (Skills). Jeder Mitarbeiter hat genau einen Vorgesetzten innerhalb des Teams; nur der Chef ist für sich selbst verantwortlich.

Jedes Projekt hat eine bestimmte Aufgabe in einer bestimmten Zeit zu erfüllen.

In dem Projekt gibt es außerdem Teams. Jedes Team besteht aus mindestens einem Mitarbeiter und hat eine bestimmte Teilaufgabe innerhalb des Projektes zu erfüllen. Es muss sichergestellt sein, dass nur Projektmitarbeiter als Teammitglieder eingesetzt werden. Eines der Teammitglieder ist der Teamleiter.

Verwenden Sie neben Elementen auch Attribute. Wo lassen sich bei Ihnen Parameter-Entitäten sinnvoll einsetzen?

- b) Geben Sie ein XML-Dokument an, das Ihre DTD erfüllt. Verwenden Sie die DTD dabei als externe DTD.

Setzen Sie an geeigneter Stelle Entitäten ein!

2.2

- a) Laden Sie vom Server die HTML-Datei `notWellformed.html` und stellen Sie sie mit einem Web-Browser dar. Die Seite sollte fehlerfrei angezeigt werden, obwohl sie im XML-Sinne nicht wohlgeformt ist.

Ändern Sie jetzt diese Datei so ab, dass Sie die Eigenschaft der Wohlgeformtheit erfüllt und die Darstellung im Browser unverändert bleibt – dabei sollen Sie *keine neue* Datei erstellen, die dieselbe Ausgabe liefert, sondern die gegebene Datei schrittweise modifizieren!

- b) Laden Sie vom Server die XML-Datei `notWellformed.xml`. Auch diese Datei ist nicht wohlgeformt. Korrigieren Sie die Fehler sinnvoll, so dass daraus eine wohlgeformte Datei entsteht.

Programmieren 3 – Grundlagen der Webprogrammierung

Aufgabenblatt 3

3.1

Laden Sie vom Server die XML-Datei `namespace.xml` und bestimmen Sie für alle enthaltenen Tags den jeweiligen Namensraum.

3.2

- a) Definieren Sie mithilfe regulärer Ausdrücke einen XML-Datentyp `distance`, in dem Entfernungsangaben gespeichert werden können.

Die Entfernungsangaben bestehen aus je einem Zahlenwert mit bis zu drei Nachkommastellen gefolgt von genau einem Leerzeichen und der anschließenden Einheit `m` oder `km`.

Bei dem Zahlenwert sollen keine redundanten führenden Nullen zugelassen sein!

Erlaubte Angaben sind z. B. `0,567 m` oder `123 km`. Nicht erlaubt sind `003,4 m`, `0,567 cm` oder `123,2231 km`.

- b) Modifizieren Sie Ihren Datentyp `distance` aus dem Aufgabenteil a so, dass der Zahlenwert vor dem Komma in Tausendergruppen durch einen Punkt gegliedert wird (Beispiel: `1.234,2 km` oder `1.234.000,2 m`).

3.3

Modellieren Sie den Sachverhalt aus Aufgabe 2.1 jetzt mithilfe eines XML-Schemas.

Definieren Sie dabei auch geeignete eigene Datentypen oder wandeln Sie die vordefinierten Datentypen für Ihre Anforderungen entsprechend ab.

Sie brauchen für Ihr Schema *keinen* expliziten Namensraum zu definieren.

Testen Sie Ihr Schema mit einer geeigneten XML-Instanz.

Programmieren 3 – Grundlagen der Webprogrammierung

Aufgabenblatt 4

4.1

Laden Sie vom Server die Datei `persons.xml`.

Erstellen Sie für die folgenden Suchanfragen bzw. Aufgaben zu diesem Dokument entsprechende XPATH-Ausdrücke und testen Sie diese:

- a) Bestimmen Sie die Namen aller Kinder der Wurzel des Dokumentenbaumes!
- b) Wie viele Personen kommen in dem Dokument vor?
- c) Welche Personen haben keinen Namen, sondern nur einen Vornamen?
- d) Welche Personen sind in dem Dokument nicht als Kinder anderer Personen aufgeführt?
- e) Finden Sie zu jedem im Dokument enthaltenen Kommentar die Person, die der *nächste* Vorfahre des Kommentarknotens ist, und geben Sie deren Vornamen aus!
- f) Welche Personen haben denselben Wohnort wie `Tieger`?
- g) Finden Sie alle Personen deren Vor- oder Nachname den Buchstaben „r“ enthält!
- h) Wie groß ist die durchschnittliche Kinderanzahl, wenn Sie für `I-Ah` auch noch zwei Kinder eintragen?
- i) Für wie viele Personen sind Wohnorte aufgeführt?
- j) Wie viele *verschiedene* Wohnorte sind aufgeführt?
- k) Wie oft ist in dem Dokument jemand als Freund einer anderer Person aufgeführt?⁴

4.2

Diese Aufgabe bezieht sich auf das XML-Dokument, dass Sie bereits in der Aufgabe 3.3 erstellt haben.

- a) Stellen Sie sicher, dass sowohl die Teams und als auch die Mitarbeiter eindeutig identifiziert werden können. Achten Sie dabei auch auf das Einhalten der referentiellen Integrität!
Setzen Sie dafür im Schema die Elemente `xs:unique`, `xs:key` oder `xs:keyref` ein.⁵
- b) Erstellen Sie für die folgenden Suchanfragen zu diesem Dokument entsprechende XPATH-Ausdrücke und testen Sie diese:
 - Wie viele Personen sind im Dokument notiert?
 - Listen Sie alle Skills der Personen auf?

⁴ schwierig!

⁵ Zur Verwendung von `xs:unique`, `xs:key` oder `xs:keyref` siehe Quellenhinweis auf den Vorlesungsfolien.

- Wie heißen die Teamleiter?
- Wie groß ist die durchschnittliche Teamgröße?
- Für welche Projekte ist die Laufzeit noch nicht abgelaufen?

Programmieren 3 – Grundlagen der Webprogrammierung

Aufgabenblatt 5

5.1

Grundlage dieser Aufgabe ist die aus Aufgabe 4.1 bekannte Datei `persons.xml`. Erweitern Sie diese XML-Datei um mehrere `children`-Elemente (einige davon auch mehrstufig!).

Schreiben Sie nun ein Java-Programm, das mithilfe von SAX diese erweiterte Datei verarbeitet. Schalten Sie dabei die Validierung ein:

- a) Geben Sie zunächst alle in der XML-Datei vorkommenden Vornamen aus.
- b) Geben Sie alle Daten genau der Personen aus, die als `children` in der Datei aufgeführt sind.
- c) Entwerfen Sie eine Klasse `Person`, in der Sie alle Angaben des in der Datei definierten Elements `person` speichern können.

Bei der Verarbeitung der XML-Datei speichern Sie jetzt unter Verwendung entsprechend vieler `Person`-Instanzen die Daten der im Aufgabenteil b gefundenen Personen in einer `Map` und geben die Daten nach der Verarbeitung der Datei nach Vornamen sortiert wieder aus.

- d) Wie müssen Sie Ihr Programm verändern, wenn nur die Personen gefunden werden sollen, die als `children` aufgeführt sind, aber selbst keine `children` besitzen?

5.2

Lösen Sie das Problem aus Aufgabe 5.1 c statt mit SAX jetzt wahlweise mithilfe der Cursor- oder Iterator-API von StAX.

Vergleichen Sie die Vor- und Nachteile der Implementierungen aus 5.1 c und 5.2?

Programmieren 3 – Grundlagen der Webprogrammierung

Aufgabenblatt 6

6.1

- a) Lösen Sie das Problem aus Aufgabe [5.1 c](#) jetzt mithilfe der DOM-API.
Worin bestehen die Vor- und Nachteile der beiden APIs?
- b) Testen Sie in Java mithilfe der DOM-API weitere XPATH-Ausdrücke aus Aufgabe [4.1](#).
Gelingt jetzt eine Lösung für die letzte der dortigen Teilaufgaben?

Programmieren 3 – Grundlagen der Webprogrammierung

Aufgabenblatt 7

7.1

Studieren Sie den ersten Teil „The JavaScript Language“ des Tutorial unter javascript.info, und vertiefen Sie damit Ihre JavaScript-Grundkenntnisse aus der Vorlesung.

7.2

- a) Entwerfen Sie eine einfache XHTML-Seite, die mehrere Instanzen der Elemente `h1`, `h2` und `p` enthält. Außerdem sollen Listen mit mehreren `li`-Instanzen vorhanden sein – auch geschachtelte Listen!
- b) Nach dem Laden der Seite in einem Browser⁶, soll auf der Seite für jedes der im Aufgabenteil [a](#) genannten Elemente automatisch ein „Button“ eingefügt werden. Wird einer dieser Buttons angeklickt, so soll die Anzahl der zugehörigen Element-Instanzen bestimmt und ebenfalls auf der Seite dargestellt werden.
- c) Fügen Sie jetzt für jedes der im Aufgabenteil [a](#) genannten Elemente einen weiteren „Button“ hinzu, der beim Anklicken sowohl das Löschen aller Instanzen des entsprechenden Elements auf der Seite als auch das Entfernen der beiden zu dem Element gehörigen Buttons auslöst.

Gestalten Sie Ihre Lösung so flexibel, dass auch noch weitere Elemente mit möglichst wenig Änderungen am Code entsprechend verarbeitet werden können.

⁶ Entsprechend den Vorgaben aus der Vorlesung werden nur neuere Browser der Mozilla- oder Firefox-Serie verwendet.