

# XML-Technologie: Übungsblatt 4

## Version 1.0

**Ausgabe am 23. November 2016, Abgabe (optional) bis zum 4. Dezember 2016**

**Vorbereitung:** Arbeiten Sie sich an Hand der Folien, des Skripts oder anderer Quellen in XQuery und auch in die XQuery Update Facility ein. Recherchieren Sie auch das Konzept eines XQuery Moduls. Schreiben Sie ein paar einfache Test-Queries und führen Sie sie aus. Beachten Sie, dass Sie in oXygen den XQuery-Prozessor Saxon explizit dafür konfigurieren müssen, dass er XQuery Updates unterstützt.

**Aufgabe 1:** In Blatt 3 haben Sie ein Datenmodell für Ihr Mancala-Spiel entwickelt, in Form eines UML Klassendiagramms. Ergänzen Sie nun das Klassendiagramm durch Methoden, die auf den Objekten der entsprechenden Klassen operieren (nicht-statische Methoden oder Objektmethoden im Sinne der objekt-orientierten Programmierung).

Typische Methoden, die Sie benötigen, sind Konstruktoren und Methoden zur Ausführung von Spielzügen. Aus diesen Methoden ergeben sich weitere "Hilfsmethoden" durch funktionale Zerlegung.

Beachten Sie, dass es möglich sein muss, von mehreren Browsern aus auf einem Server mehrere Mancala-Spiele gleichzeitig zu spielen.

Um zu einem vollständigen Modell zu kommen, beschreiben Sie einige Spielverläufe und auch Interleavings von mehreren Spielverläufen als Use Cases. Sie können auch Aktivitätsdiagramme anlegen, um die Aufruf-Hierarchie Ihrer Methoden zu modellieren. Auf äußerster Ebene erfolgen die Aufrufe von Methoden als Reaktion auf Requests vom Web Browser.

Dokumentieren Sie Ihr Design in DocBook.

**Aufgabe 2:** Setzen Sie Ihr Design von Aufgabe 1 als XQuery-Modul um. Eine nicht-statische Methode oder Objektmethode realisieren Sie als XQuery-Funktion, der Sie die Daten des Objekts als zusätzlichen Parameter (in der Rolle von "self" oder "this") mitgeben. Änderungen, die die Methode an den Daten des Objekts vornimmt, sollen dann in der XML-Repräsentation des Objekts gespeichert werden.

Zur Umsetzung müssen Sie sich ein Konzept überlegen, wie Sie Objekte und ihre XML-kodierten Daten referenzieren. Hier können Sie IDs benutzen, die Sie beim Kreieren der Objekte dynamisch generieren.

Sie müssen sich auch überlegen, wie Sie die Daten Ihrer Objekte speichern wollen, entweder in einer Container-Datei oder in einzelnen Dateien pro Objekt. Ihre Design-Entscheidung hat Konsequenzen dafür, wie Sie referenzierte Objekte zugreifen.

Sie brauchen sich keine Gedanken über die Aufruffreihenfolge Ihrer Methoden zu machen; d.h. Sie können davon ausgehen, dass zu jedem Zeitpunkt nur Methoden aufgerufen werden, die semantisch Sinn machen. Die Aufruffreihenfolgen (das Verhalten des Systems) sichern wir in der nächsten Ausbaustufe des Projekts ab.

Ebenso brauchen Sie sich keine Gedanken über Concurrency / Race Conditions / Atomarität zu machen. D.h. Sie können davon ausgehen, dass jede Ihrer XQuery-Methoden vollständig abgearbeitet wird, bevor eine andere Query aufgerufen wird. Nebenläufigkeit sichern wir in der nächsten Ausbaustufe des Projekts durch Einsatz einer XML-Datenbank ab, die die Queries ausführt.

Eine Funktion in XQuery kann entweder XML-Daten als Ergebnis zurückliefern oder Updates vornehmen, aber nicht beides. Auch die Komponierbarkeit von XQuery-Funktionen mit Updates ist eingeschränkt. Entwerfen Sie Ihre Methoden zunächst in einer Form, die dieses Probleme umgeht. Wir werden das Problem mit HTTP Redirects lösen, wenn wir XQuery serverseitig ausführen.

Dokumentieren Sie Ihre Lösung in DocBook.

**Präsentation:** Zwei Gruppen können ihre Lösung zu Aufgabenblatt 4 im Praktikum am 30. November präsentieren und dazu einen Notenbonus erhalten. Es gelten die folgenden Spielregeln:

1. Die Gruppen, die sich zuerst per E-Mail bei mir melden, bekommen den Zuschlag.
2. Es gilt ein striktes Zeitlimit von 3 Minuten pro Gruppenmitglied für die Präsentation.
3. Die Lösung muss 36 Stunden vor der Präsentation per E-Mail vorzeitig abgegeben werden.
4. Nur Studierende, die sich an der Präsentation aktiv beteiligen, bekommen den Notenbonus.
5. Konzentrieren Sie sich in der Präsentation auf die Organisation Ihrer Lösung, wie in der Aufgabenbeschreibung angegeben.
6. Sie können zentrale Aspekte Ihrer Implementierung auch im Code zeigen. Aber bitte keine unstrukturierten Walk-throughs durch Code.

**Abgabe:** Geben Sie Ihre Lösung bis zum 4. Dezember 2016 in einem Archiv-Format in Moodle ab. Geben Sie Ihrer Archiv-Datei einen unverwechselbaren Namen, z.B. ein Kürzel aus den Namen Ihrer Gruppenmitglieder. Schreiben Sie außerdem Ihre Namen als Kommentar in jede einzelne Datei.

**Leistungsstand:** Die folgenden Lernziele sollten Sie inzwischen erreicht haben:

- Verständnis für den Hintergrund von XML.
- Beherrschen der Sprache XML: Instanzen mit Elementen, Attributen, Referenzen und Text; Definition von Entitäten in DTD.
- Parsen von XML-Dokumenten.
- Visualisieren von XML-Dokumenten in Web-Browsern mit CSS.
- Einsatz von Kodierungsformaten in XML.
- Einsatz von Namensräumen in XML.
- Umgang mit DocBook.
- Strukturierung und Implementierung von SVG-Dokumenten.
- Weitere Vertrautheit mit XML DTD; Modellierung von XML-Anwendungen; Deklaration von Elementen und Attributen.
- Praktischer Umgang mit XPath.
- Entwicklung modularer und konfigurierbarer XSLT-Programme.

**Lernziele:** Konkrete Lernziele für die Arbeit in der kommenden Woche sind:

- Formulierung von XQuery-Abfragen und XQuery-Updates.
- Definition und Aufruf von XQuery-Funktionen in Modulen.
- Umsetzung von Objektmethoden mit XQuery-Funktionen auf XML-kodierten Daten.

**Ausblick:** Zu folgenden Themen wird es noch Technologieeinführungen geben (maximal zwei):

- Benutzen von einer XML-Datenbank (BaseX) als XQuery-Prozessor.
- GUIs mit XForms
- Architektur von Web-Anwendungen: der X-Stack

Es gibt maximal noch ein weiteres Aufgabenblatt.