

## Übung „Netzbasierte Informationssysteme“ WS 2008/2009

### Übungsblatt 3

Ausgabe am 11.11.2008

Abgabe bis spätestens 18.11.2008, 16.00 Uhr

#### Aufgabe 8 – Modelling XML CustomerDB (4 Punkte):

Im Rahmen eines Industrie-Praktikums werden Sie mit dem Design einer einfachen XML-basierten Kundendatenbank beauftragt. Ein Mitarbeiter der CRM-Abteilung gibt Ihnen hierfür folgende Informationen mit auf den Weg:

*Ein Kunde hat genau einen Nachnamen und einen Vornamen sowie eine eindeutige Kundennummer. Ein Kunde hat eine oder mehrere Wohnungen, wobei eine Wohnung über die Eigenschaften Straße, Hausnummer, PLZ und Wohnort eindeutig identifizierbar ist. Einige Kunden besitzen eine Emailadresse, alle haben eine Telefonnummer.*

Die Kundendaten sollen in einem XML-basierten Format zwischengespeichert werden.

- Entwerfen Sie eine DTD `aufgabe8_customerDB.dtd` für solch einen XML-basierten Kundendatenspeicher. Es müssen alle oben angegebenen Informationen in XML-Form speicherbar sein. Insbesondere soll in dieser Spezifikation ein Kunde unbegrenzt viele Wohnungen haben, mindestens jedoch eine. Achten Sie auf Pflichtteile sowie Teile, die optional sind! Insbesondere soll es auch möglich sein, keinen Kunden in einer validen XML-Datei enthalten zu haben (leerer Kundenspeicher).
- Durch die soeben spezifizierte DTD wird ein bestimmtes XML-Format beschrieben. Legen Sie nun eine valide Instanz davon an. Diese Instanz (`aufgabe8_customerDB.xml`) soll folgende Kundendaten enthalten:

Name	Meier	Spatz	Osterwelle
Vorname	Karl	Michaela	Guido
Telefon	0821654321	01727654321	01797654321
Emailadresse	k.m@oberkunde.de	adler@inf.fu-berlin.de	-
Straße, Hausnr.	Fichtelstraße, 4	Buchegger Platz, 4	Hauptstraße, 12
PLZ	86165	48234	48123
Wohnort	Augsburg	Muenster	Muenster
		Zweitwohnsitz:	
		Am Bergweg, 4	
		86842 Irsingen	

Eine Zeile in Ihrem File entspricht dabei einem einzelnen Kundendatensatz.

- Überprüfen Sie die Validität Ihrer Instanz gegen die DTD indem Sie in einem zweiten File `customerDB_inline.xml` zusätzlich zu den Daten die DTD inline angeben und einen Online-Validator wie z.B. unter <http://www.validome.org/xml/> nutzen.
- Entwerfen Sie nun ein XML-Schema das funktional der entwickelten DTD entspricht (`aufgabe8_customerDB.xsd`).

#### Aufgabe 9 – Offline CustomerDB: Show und Insert (6 Punkte):

In Aufgabe 8 haben Sie das XML-basierte Format zur Speicherung von Daten des Kundendatenbanksystems entworfen. Ziel dieser Aufgabe ist es, mittels Webanwendungen Zugriff auf den Inhalt der Kundendatenbank zu erhalten. Entwickeln Sie dazu folgende Komponenten:

- Ein JSP namens `ShowContent.jsp` (URL: `/aufgabe9/ShowContent`) gibt den Inhalt der Kundendatenbank aus einem File `aufgabe9_customerDB.xml` (im Tomcat Verzeichnis) - in übersichtlicher, tabellarischer Form - aus. Verschiedene Adressen eines Kunden sollten diesem eindeutig zugeordnet werden können. Falls noch keine Kundeninstanz angelegt wurde, sollte eine entsprechende Meldung zum Client zurückgesendet werden. Sie können bei dem übergebenen File davon ausgehen, dass es der DTD bzw. dem Schema aus Aufgabe 8 entspricht, also valide ist. Verwenden Sie einen SAX- oder DOM-Parser. Falls keine Kundendaten enthalten sind, soll `ShowContent` leere Tabellen anzeigen. Deployen Sie das JSP unter `/aufgabe9/showContent`. Testen Sie die Funktionalität Ihres JSPs indem Sie den Inhalt Ihrer Kundeninstanz aus Aufgabe 8 in Ihren neuen XML-Speicher kopieren und mittels `ShowContent` anzeigen lassen.

- Nun sollen auch neu eingegebene Kundendatensätze im XML-Datenspeicher abgelegt werden können. Ein Servlet CreateCustomer (URL: /aufgabe9/createCustomer) generiert ein HTML-Formular, über welches ein User neue Kundendaten in die Datenbank (in die Datei aufgabe9\_customerDB.xml) eintragen kann. Der User soll über Erfolg oder Misserfolg der Neukundenerfassung in Kenntnis gesetzt werden. Bei Erfolg der Erfassung soll dem Kunden zusätzlich seine neue Kundennummer mitgeteilt werden. Eine Zeile im file soll dabei einem kompletten Kundendatensatz entsprechen. Bereits in der Kundendatei enthaltene Daten werden nicht gelöscht. Der neue Kundendatensatz wird einfach nach den bestehenden Daten in die Datei geschrieben. Der Einfachheit halber können Sie für das Formular davon ausgehen, dass kein Kunde mehr als zwei verschiedene Wohnungen besitzt.

#### Aufgabe 10 – XPath Ausdrücke (3 Punkte):

Nach der erfolgreichen Einführung der XML Kundendatenbank wird diese nun intensiv sowohl für die interne als auch für die externe Kommunikation genutzt. Die Firmenleitung ist von den Möglichkeiten von XML begeistert und möchte weitere Innovationen auf diesem Gebiet einführen. Ein externer Partner schickt für die Durchführung einer Marketingkampagne ein XML-Dokument premiumCustomers.xml (siehe NBI-Homepage), welches eine Reihe von Kundendaten enthält. Die Firmenleitung beauftragt Sie mit der Selektion der für die Kampagne wichtigen Daten. In der CSW Netzbaasierte Informationssysteme-Vorlesung haben Sie während Ihres Studiums XPath kennengelernt. Sie erkennen sofort, dass sich diese Technologie für diese Aufgabe bestens eignet.

1) Geben Sie jeweils einen XPath-Ausdruck an, der folgende Knotenmengen in premiumCustomers.xml lokalisiert:

2. Kunde-Knoten des Dokuments.
- Alle Kunden, die im Postleitzahlengebiet PLZ = 48234 wohnen.
- Den ersten Kunde-Knoten, für welches das Attribute „geschlecht“ ungleich „M“ ist.
- Alle Kunden, die 1972 geboren sind.
- Die Vornamen aller Kunden die *Schmidt* mit Nachnamen heißen.
- Die Kunden-Elemente deren einziges Hobby *Konzert* ist.
- Den Wohnort der Kunden, die vor 1970 geboren wurden.
- Die Kundennummer aller Kunden, die Golf spielen.

2) Welche Knotenmengen aus premiumCustomers.xml liefern die folgenden XPath-Ausdrücke:

- /Kunden/Kunde[3]/@nachname
- //PLZ[starts-with(., "48")]
- //Kunde[@nachname = preceding::Kunde/@nachname or @nachname = following::Kunde/@nachname]
- //Kunde[@geburtsjahr < sum(//\*/@geburtsjahr) div 5]
- //Kunde[(position() mod 2)=0]/@geschlecht

Geben Sie ihre Lösungen strukturiert in einem Textfile aufgabe10\_XPath.txt ab.

#### Aufgabe 11 - XQuery (2 Punkte):

Geben Sie eine XQuery-Anfrage an, die basierend auf premiumCustomers.xml alle in Augsburg lebenden Kunden unter Angabe der Kundennummer, des Nachnamens und des Hobbys in einer XML-Liste der Form

```
<result>
  <kundennr> Kundennummer </kundennr>
  <name> Nachname </name>
  <hobby> Hobby </hobby>
</result>
```

ausgibt. Geben Sie ihre Lösung in einem Textfile aufgabe11\_XQuery.txt ab.

#### Bonus-Aufgabe 12 – XSLT: UltimateConverter (5 Bonus-Punkte):

Die Firmenleitung wünscht sich, direkt online – und damit plattformunabhängig - Einsicht in Kundendateien nehmen zu können. Implementieren Sie mittels XSLT und Xalan einen XML to HTML Converter namens UltimateConverter.java für XML-files, die dem Aufbau von premiumCustomers.xml entsprechen. Orientieren Sie sich dabei an den Beispielen aus den Übungsfolien. Die HTML-Ausgabe soll übersichtlich und in Tabellenform sein. Der Aufruf des Converters soll lokal im Eclipse-Workspace erfolgen, das XML-Quellfile soll direkt im entsprechenden Pfad liegen. Testen Sie die Syntax Ihrer HTML-Ausgabe indem Sie premiumCustomers.xml nach premiumCustomers.html konvertieren und das Ergebnis in Ihrem Tomcat unter /aufgabe12 deployen.

Viel Erfolg!