

# PI.SWA.SA.VO Software Architekturen und Web Technologien

Schriftliche Einzelprüfung zur Vorlesung des Wintersemesters 2008/09

Dritter Prüfungstermin

Di 28.04.2009

Prüfer: Dr. Michael Derntl

## Anweisungen:

- **Arbeitszeit: 90 Minuten**, es sind **max. 90 Punkte** zu erreichen (die Anzahl der Punkte, die pro Aufgabe zu erreichen ist, finden Sie jeweils in eckiger Klammer). Kalkulieren Sie also für jeden erreichbaren Punkt ca. eine Minute Arbeitszeit.
- Halten Sie einen **Lichtbildausweis** (Studentenausweis, Führerschein, Pass, etc.) bereit.
- Lösen Sie die Aufgaben direkt auf den Angabebblättern und schreiben Sie **leserlich**.
- Falls Sie zusätzliche Blätter benötigen, fragen Sie die Prüfungsaufsicht.
- Es sind **keine Unterlagen**, Handies, PDA, Laptops, Taschenrechner, Headsets, oder sonstige Hilfsmittel erlaubt!

## Hinweise:

- **Ergebnisse** werden bis spätestens **Di 12.5.** im PISWI verfügbar sein.
- **Einsicht** ist **nach Terminvereinbarung** per E-Mail. Spätester Termin: **Mo 18.5.**
- Der letzte Prüfungstermin für das Stoffsemester WS 2008/09 wird Anfang Juni stattfinden.

**Aufgabe 1: XML Technologien****[41]**

Gegeben ist folgendes XML Schema. (Hinweis: die Zahlen am Beginn jeder Zeile sind Zeilennummern, auf die Sie sich bei der Lösung der Aufgabe beziehen können.)

```
1 <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
2
3   <xsd:complexType name="wohnungen_t">
4     <xsd:sequence minOccurs="2" maxOccurs="unbounded">
5       <xsd:element name="wohnung" type="wohnung_t" minOccurs="0"/>
6       <xsd:element name="makler" type="xsd:string" minOccurs="0"/>
7     </xsd:sequence>
8   </xsd:complexType>
9
10  <xsd:complexType name="wohnung_t">
11    <xsd:sequence>
12      <xsd:element name="eigentuemer" type="eigentuemer_t"/>
13      <xsd:element name="mieter" type="xsd:string"/>
14      <xsd:element name="typ" type="typ_t"/>
15    </xsd:sequence>
16    <xsd:attribute name="ort" type="xsd:string" use="required"/>
17    <xsd:attribute name="flaeche" type="xsd:positiveInteger" use="required"/>
18    <xsd:attribute name="info" type="xsd:string" use="optional"/>
19  </xsd:complexType>
20
21  <xsd:simpleType name="eigentuemer_t">
22    <xsd:restriction base="xsd:string">
23      <xsd:maxLength value="10"/>
24    </xsd:restriction>
25  </xsd:simpleType>
26
27  <xsd:simpleType name="typ_t">
28    <xsd:restriction base="xsd:string">
29      <xsd:enumeration value="Miete"/>
30      <xsd:enumeration value="Eigentum"/>
31      <xsd:enumeration value="Geschäft"/>
32      <xsd:enumeration value="Pacht"/>
33    </xsd:restriction>
34  </xsd:simpleType>
35
36  <xsd:element name="wohnungen" type="wohnungen_t">
37    <xsd:unique name="x">
38      <xsd:selector xpath="wohnung"/>
39      <xsd:field xpath="eigentuemer"/>
40    </xsd:unique>
41  </xsd:element>
42
43 </xsd:schema>
```

- 1a. Schema-Validierung.** Markieren Sie jene Stellen im folgenden XML Dokument, die in Bezug auf das gegebene XML Schema *ungültig* sind. Begründen Sie jeweils kurz, warum die Stelle ungültig ist. [15]

```
1  <?xml version="1.0" encoding="utf-8"?>
2
3  <wohnungen>
4    <makler>Immobilienbüro Dr. Rohdiamant</makler>
5
6    <wohnung ort="Gols" info="möbliert">
7      <eigentuemer>«VORNAME» «NACHNAME»</eigentuemer>
8      <mieter>-</mieter>
9      <typ>Geschäft</typ>
10     <typ>Pacht</typ>
11     <zimmer>7</zimmer>
12   </wohnung>
13
14   <wohnung ort="Wien 17" flaeche="92">
15     <eigentuemer>M. Derntl</eigentuemer>
16     <mieter>H. Reisinger</mieter>
17     <typ>Miete Geschäft</typ>
18   </wohnung>
19
20   <wohnung ort="Linz" flaeche="152" info="Altbaujuwel"/>
21
22   <wohnung ort="Graz" wohnflaeche="68">
23     <eigentuemer>A. Schwarzenegger</eigentuemer>
24     <typ>Geschäft</typ>
25     <mieter/>
26   </wohnung>
27
28   <wohnung ort="Wien 21" flaeche="98m²">
29     <eigentuemer>M. Derntl</eigentuemer>
30     <mieter>H. Hölzl</mieter>
31     <typ>Pacht</typ>
32   </wohnung>
33 </wohnungen>
```

**1b. Schema vs. DTD.** Erklären Sie, welche in dem gegebenen XML Schema definierten Einschränkungen Sie mit einer DTD nicht definieren könnten (mit kurzer Begründung!) [6]

**1c. XPath.** Nehmen Sie an, es existiert ein XML Dokument, das gültig für das gegebene XML Schema ist. Erstellen Sie nun je einen XPath Ausdruck für folgende Abfragen:

i. Die Anzahl der Wohnungen mit Typ "Eigentum" außerhalb Wiens und einer Fläche von mindestens 85. [3]

ii. Die Mieter aller Wohnungen, deren Eigentümer "«VORNAME» «NACHNAME»" heißt. [2]

iii. Alle Orte, in denen es mindestens zwei Wohnungen vom Typ „Miete“ gibt. (Hinweis: dafür brauchen Sie XPath Achsen!) [3]

**1d. XSL Transformation.** Gegeben ist folgende DTD:

```
<!ELEMENT personen (person*)>
<!ELEMENT person (email?, homepage*)>
<!ATTLIST person id ID #REQUIRED vorname CDATA #REQUIRED nachname CDATA #REQUIRED>
<!ELEMENT email (#PCDATA)>
<!ELEMENT homepage (#PCDATA)>
<!ATTLIST homepage typ (privat|beruflich) "privat">
```

Gesucht ist ein XSL-Stylesheet, das eine ungeordnete HTML-Liste (<UL>) aller Personen (Vor- und Nachname) ausgibt, die über eine E-Mail-Adresse verfügen. Zu jeder Person soll auch in Klammer die E-Mail-Adresse (als Hyperlink mit `mailto:`) ausgegeben werden. [12]

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"/>
  <xsl:output method="html"/>
```

```
</xsl:stylesheet>
```

## Aufgabe 2: Software Architektur

[30]

**2a. Architekturprinzipien.** Beantworten Sie jeweils kurz folgende Fragen:

i. Was versteht man unter Kopplung?

[2]

ii. Welche Vorteile bietet eine lose Kopplung zwischen Komponenten und wie hängt eine lose Kopplung mit der Kohäsion zusammen?

[3]

iii. Was ist eine Schnittstelle und welche Zwecke erfüllt sie in einer Software Architektur? Welche Auswirkungen hat die Verwendung von Schnittstellen auf Kopplung und Kohäsion?

[3]

**2b. Architekturpatterns.**

i. Erläutern Sie kurz den Zweck des Broker Pattern und wann er eingesetzt wird. [2]

ii. Skizzieren Sie die Broker Pattern Komponenten und deren Beziehungen, und erklären Sie welche Aufgaben die Komponenten jeweils erfüllen. [8]

iii. Beschreiben Sie kurz die Vor- und Nachteile des Einsatzes einer Broker Pattern Lösung. [3]

**2c. Anforderungen.** Was versteht man unter nichtfunktionalen Anforderungen (NFR)? Welche Arten von NFR gibt es? Schlagen Sie drei NFR für ein Online Lehrveranstaltungs-Anmeldesystem vor. [9]



### Aufgabe 3: Webservices

[19]

3a. Die Operationen eines Prüfungsverwaltungs-Webservices sind wie folgt definiert:

```
noteEintragen(  
    pruefung: Prüfung,  
    teilnehmer: Student,  
    note: Ganzzahl) : Boolean  
  
allePrüfungenVonStudent (  
    matrNr: String) : Prüfung[]
```

Erklären Sie verbal, was beim Erstellen der WSDL Datei für dieses Webservice zu tun wäre.  
Gehen Sie dabei konkret auf alle Abschnitte der WSDL Datei ein!

[12]

**3b.** Was ist UDDI und wozu wird es im Zusammenhang mit Webservices verwendet? [3]

**3c.** Wozu wird ein SOAP Envelope verwendet und was ist darin definiert? [4]