

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

Schriftliche Einzelprüfung zur Vorlesung des Wintersemesters 2008/09

Erster Prüfungstermin

Do 22.01.2009

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 Angabeblättern 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 **So 8.2.** im PISWI verfügbar sein.
- **Einsicht** ist **nach Terminvereinbarung** per E-Mail bis spätestens **Mi 18.2.**
- Drei weitere Prüfungstermine werden im Sommersemester 2009 folgen.

Aufgabe 1: XML Technologien

[38]

Gegeben ist folgende XML Datei:

```
<?xml version="1.0" encoding="utf-8"?>
<ehopaare>
  <ehopaar hochzeitsTag="1983-01-22">
    <scheidung>2001-12-24</scheidung>
    <mann svnr="2948120145">Karli Zwiefel</mann>
    <frau svnr="1234050342">Mitzi Knofel</frau>
    <standesamt>Wien 9</standesamt>
  </ehopaar>
  <ehopaar hochzeitsTag="2002-08-12">
    <frau svnr="1234050342">Mitzi Knofel</frau>
    <mann svnr="3203070382">Franz Radiwurzn</mann>
    <standesamt>St. Pölten</standesamt>
  </ehopaar>
  <ehopaar hochzeitsTag="1954-09-02">
    <standesamt>Los Angeles</standesamt>
    <frau svnr="4419021232">Jane Doe</frau>
    <mann svnr="1934081220">Henry Chinaski</mann>
    <scheidung>1957-01-01</scheidung>
  </ehopaar>
  <ehopaar hochzeitsTag="1998-02-02">
    <mann svnr="4756260268">Ihsan Farha</mann>
    <frau svnr="3487010157">Chip Stack</frau>
    <standesamt>Las Vegas</standesamt>
  </ehopaar>

  ... weitere Ehepaare ...

</ehopaare>
```

1a. XPath. Formulieren Sie jeweils einen XPath für folgende Abfragen auf die obige XML Datei:

i. Die Anzahl der am Standesamt St. Pölten geschlossenen Ehen. [2]

ii. Alle Standesämter, an denen bereits Ehen geschlossen wurden, jedoch nur solche, die nicht geschieden wurden. [4]

iii. Die Namen aller Frauen, die bereits mehrmals verheiratet waren. Anmerkung: Die Sozialversicherungen (Attribut **svnr**) seien eindeutig. [4]

iv. Die Scheidungsrate (also das Verhältnis von geschiedenen Ehen zu nicht geschiedenen Ehen). [3]

- 1b. XSLT.** Sie wollen nun die obige XML Datei nach HTML transformieren, sodass der Output wie im folgenden Ausschnitt angedeutet aussehen soll: [12]

```
<ol>
  <li>Karli Zwiefel heiratet Mitzi Knofel am 1983-01-22 im Standesamt Wien 9.</li>
  <li>Franz Radiwurzn heiratet Mitzi Knofel am 2002-08-12 im Standesamt St. Pölten.</li>
  ... und so weiter ...
</ol>
```

Fügen Sie das benötigte Template in folgendes Gerüst ein:

```
<?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>
```

1c. XML Schema. Komplettieren Sie die grau unterlegten Teile des unten stehenden Schemas für die gegebene XML Datei, wobei neben den in der XML Datei ersichtlichen Einschränkungen folgende Zusatzangaben gelten. [13]

- Die Sozialversicherungsnummer (Attribut `svnr`) ist ein String mit genau 10 Zeichen.
- Der Hochzeitstag (Attribut `hochzeitsTag`) ist ein Datum.
- Es gibt keine optionalen Attribute.

```

<?xml version="1.0" encoding="utf-8"?>
<xsd:schema xmlns:  "http://www.w3.org/2001/XMLSchema">
  <xsd:element name="ehopaare" type="wurzelType"/>

  <xsd:complexType name="wurzelType">
    <xsd:sequence>
      <xsd:element name="  " type="ehopaarType" maxOccurs="  "/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="personType">
    <xsd:  >
      <xsd:extension base="xsd:string">
        <xsd:attribute type="svnrType"  />
      </xsd:extension>
    </xsd:  >
  </xsd:complexType>

  <xsd:simpleType name="svnrType">
    
  </xsd:simpleType>

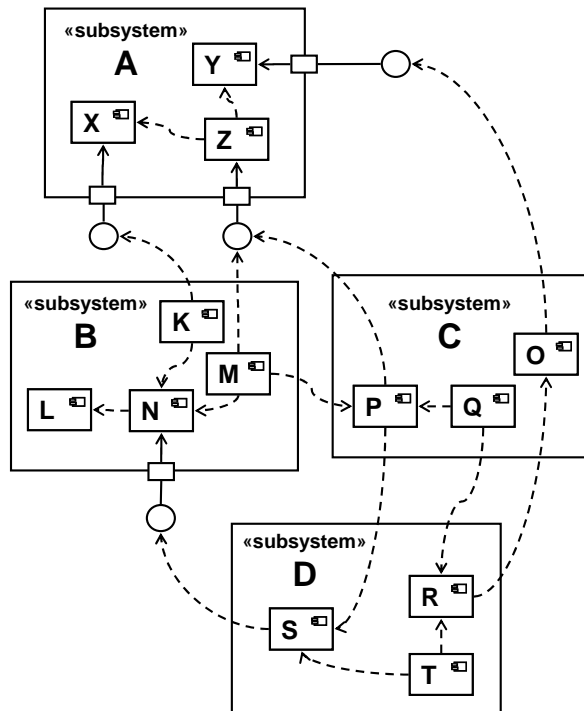
  <xsd:complexType name="ehopaarType">
    
  </xsd:complexType>
</xsd:schema>

```

Aufgabe 2: Software Architektur

[32]

- 2a. **Architekturprinzipien.** Beurteilen Sie das Ausmaß von Kohäsion, Kopplung und der Verwendung von Schnittstellen der Subsysteme A, B, C und D im folgenden Architekturmodell (mit kurzer Begründung jeweils). [10]



2b. SWA Dimensionen. Nennen und charakterisieren Sie kurz die sechs Dimensionen des SWA Ordnungsrahmens (*Hinweis: Die Dimensionen beschäftigen sich mit den Fragen: Was? Wo? Womit? Warum? Wer? Wie?*) [12]

2c. Architektursichten und -ebenen.

- i. Was versteht man unter einer Architektursicht und wie hängen Architektursichten mit Architekturebenen zusammen? [3]

- ii. Nennen und beschreiben Sie mindestens drei der fünf Sichten des *Abstrakten Architektur Modells*. [7]

Aufgabe 3: Webanwendungen

[20]

3a. Representational State Transfer (REST).

- i. Erläutern Sie Konzept und Eigenschaften einer REST Architektur anhand eines selbstgewählten konkreten Beispiels. [8]

- ii. Welche Vorteile bieten REST Architekturen?

[3]

3b. Sessions und Cookies.

- i. Was versteht man unter Sessions und Cookies und wozu werden sie in Webanwendungen verwendet/benötigt? [5]

- ii. Geben Sie konkrete sinnvolle Beispiele für Cookie- und Sessiondaten in einer Webanwendung für einen Online-Buchshop. [4]