

Übungsblatt Klausurvorbereitung

Die Aufgaben werden in der Übung am 22.7. um 11:00 in (wird noch bekannt gegeben), soweit möglich besprochen. Präferenzen für bestimmte Aufgaben bitte am Start der Übung angeben.

Dies ist *keine* Probeklausur sondern soll der Vertiefung des Wissens dienen.

Aufgabe 1 : Network Protocols: Multiple Choice

Kreuzen Sie Zutreffendes an:

(a) IP-Adressen (V4) besitzen folgende Eigenschaften:

- Sie bestehen aus 4 Byte.
- Sie sind logisch in drei Teile gegliedert: Adresspräfix, Adressinfix und Adresssuffix.
- Der Adresssuffix identifiziert einen Host-Rechner.

(b) Prinzipien des Portkonzepts sind:

- Ein Port ist ein Dienstzugriffspunkt der Transportschicht des TCP/IP-Protokolls.
- UDP-Ports dienen zur Einrichtung eines verbindungsorientierten Transportdienstes.
- Reservierte Ports (Nummern zwischen 0 und 1023) sind weltweit eindeutig.

(c) Sie fordern über Ihren Browser die URL `http://webis1.uni-weimar.de:8080` des Servers `webis1.uni-weimar.de` an. Welche der folgenden Aussagen treffen zu?

- Ihr Browser öffnet lokal den Port 8080 und verbindet sich mit dem Server auf Port 80.
- Ihr Browser öffnet lokal den Port 8080 und verbindet sich mit dem Server auf Port 8080.
- Ihr Browser öffnet lokal einen Port und verbindet sich mit dem Server auf Port 80.
- Ihr Browser öffnet lokal einen Port und verbindet sich mit dem Server auf Port 8080.
- Ihr Browser öffnet lokal den Port 80 und verbindet sich mit dem Server auf Port 8080.
- Die angegebene URL ist falsch; HTTP funktioniert nur auf Port 80.

(d) Folgende Zusammenhänge gelten für POST bzw. GET:

- Jede Anfrage, die sich mittels POST abwickeln lässt, kann man auch durch GET realisieren.
- Jede Anfrage, die sich mittels GET abwickeln lässt, kann man auch durch POST realisieren.
- Bei einer Anfrage mittels GET sind die Parameter als Teil der URL codiert.
- Die Realisierung von Anfragen mittels GET verhindert den Einsatz von Cookies.

(e) Welche der folgenden Aussagen zu Content-Negotiation sind wahr?

- Content-Negotiation ist sinnvoll, falls Web-Dokumente in mehreren Varianten vorliegen.
- Content-Negotiation kann server-driven und agent-driven realisiert werden.
- Content-Negotiation benötigt immer einen Proxy-Server.

(f) Session-Management kann mit folgenden Techniken implementiert werden:

- URL-Rewriting
- URL-Reparsing
- Radio Buttons
- Hidden Fields

Beachten Sie, dass zu einer Frage mehrere Antworten zutreffen können. Eine Frage gilt als richtig beantwortet, falls alle zutreffenden und keine unzutreffende Antwort angekreuzt ist.

Aufgabe 2 : CSS: Pseudoklassen

Laden Sie die Datei `css-pseudoclasses.html` von der Homepage der Vorlesung.

Geben Sie jeweils ein CSS-Fragment an, um die folgenden Effekte zu erzielen. Verändern Sie dabei nicht den body des HTML-Dokuments.

- (a) Der erste Paragraph (und nur dieser) soll in blauer Schrift dargestellt werden.
- (b) Die Text hervorhebungen (``) im ersten Paragraph (und nur in diesem) sollen in roter Schrift dargestellt werden.
- (c) In beiden Paragraphen soll jeweils die erste Text hervorhebung (``) in grüner Schrift dargestellt werden. Jeder andere Text soll unverändert dargestellt werden.

Aufgabe 3 : CSS: Cascading Sizing

Laden Sie die Datei `css-size-inheritance.html` von der Homepage der Vorlesung.

- (a) Ergänzen Sie eine CSS-Regel im Header des Dokumentes, die der Schriftgröße (Property `font-size`) von Listenelementen den Wert `1.2em` zuweist. Vergleichen Sie die Darstellungen mit und ohne diese Regel in Ihrem Browser. Welches unerwünschte Verhalten tritt zu Tage und warum? Wie würde sich eine noch tiefer verschachtelte Liste verhalten?
- (b) Ändern Sie lediglich die Maßeinheit der Schriftgröße, um das unerwünschte Verhalten zu unterbinden. Welche Maßeinheit bietet sich hierfür an? Begründen Sie! Konsultieren Sie hierzu [W3C: CSS Values and Units](#).
- (c) Wie können Sie das unerwünschte Verhalten durch Einfügen weiterer Regeln verhindern?

Aufgabe 4 : XML Schema

Modellieren Sie beliebig große Matrizen mit Elementen aus den ganzen Zahlen mit Hilfe von XML Schema. Geben Sie zu ihrer XML-Schema-Datei ein XML-Instanz-Dokument an. Validieren Sie ihre XML-Dokumente.

(Fortsetzung auf nächster Seite)

Aufgabe 5 : XML Schema: Namensräume in XML-Dokumenten (1+1 Punkte)

Gegeben sei ein XML-Schema für Notizen (note-qua.xsd):

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.uni-weimar.de/note"
  elementFormDefault="qualified">
  <xsd:element name="note">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="to" type="xsd:string"/>
        <xsd:element name="from" type="xsd:string" />
        <xsd:element name="heading" type="xsd:string"/>
        <xsd:element name="body" type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Das folgende XML-Dokument beschreibt eine Notiz gemäß des obigen XML-Schemas (note-ns.xml).

```
<?xml version="1.0"?>
<a:note xmlns:a="http://www.uni-weimar.de/note"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.uni-weimar.de/note note-qua.xsd">
  <a:to>Tove</a:to>
  <a:from>Jani</a:from>
  <a:heading>Reminder</a:heading>
  <a:body>Do not forget me this weekend!</a:body>
</a:note>
```

(a) Wie ist das Instanzdokument zu ändern, wenn es mit den nachfolgenden Zeilen beginnt?

```
<?xml version="1.0"?>
<note xmlns="http://www.uni-weimar.de/note"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.uni-weimar.de/note note-qua.xsd">
  ...
```

(b) Wie ist das Instanzdokument zu ändern, wenn das XML-Schema mit den nachfolgenden Zeilen beginnt?

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.uni-weimar.de/note"
  elementFormDefault="unqualified">
  ...
```

Aufgabe 6 : XSLT: Peanuts

In der Datei `peanuts.xml` sind einige Namen und weitere Informationen enthalten. Stellen Sie die Informationen aus der Datei als HTML-Dokument dar. Das HTML-Dokument soll die Überschrift „Darsteller der Peanuts“ enthalten, sowie eine dreispaltige Tabelle mit den Spaltennamen „Vorname“, „Name“ und „Einführungsjahr“. Erstellen Sie ein eigenes XSLT-Stylesheet für jede Unteraufgabe. Nutzen Sie `peanuts-scaffold.xsl` als Grundlage.

(a) Geben Sie ein XSLT-Stylesheet an, dass die Tabelle zusätzlich nach Nachnamen sortiert.

(b) Geben Sie ein XSLT-Stylesheet an, in der die Tabelle nur Darsteller enthält, die nach dem Jahr 1960 eingeführt wurden.

Aufgabe 7 : Document Languages: Multiple Choice

Kreuzen Sie Zutreffendes an:

- (a) Welche der folgenden Aussagen aus dem Gebiet der Dokumentsprachen sind wahr?
- HTML-Dokumente sind XML-Dokumente.
 - XSL-Stylesheets sind XML-Dokumente.
 - XML-Schemata sind XML-Dokumente.
- (b) Folgende Zusammenhänge bestehen zwischen der Wohlgeformtheit und der Validität von XML-Dokumenten:
- Aus der Wohlgeformtheit eines XML-Dokumentes folgt seine Validität.
 - Aus der Wohlgeformtheit eines XML-Dokumentes läßt sich nichts über seine Validität folgern.
 - Wohlgeformtheit und Validität können nicht gleichzeitig erfüllt sein.
- (c) Welche der folgenden Eigenschaften müssen bei einem wohlgeformten XML-Dokument erfüllt sein?
- Es gibt genau ein Wurzelement.
 - Es gibt höchstens ein Wurzelement
 - Es muß UTF8-Codierung besitzen.
- (d) Bei der Verarbeitung von XML-Dokumenten mit XSL-Stylesheets gilt: Eine Template-Regel wird für einen Knoten v nicht angewandt, wenn sie für einen über v liegenden Knoten angewandt wurde.
- stimmt immer
 - stimmt, falls nur *ein* Preorder-Durchlauf stattfindet
 - stimmt nicht immer
- (e) Folgende Eigenschaften treffen auf einen XSLT-Stylesheet-Prozessor zu:
- Eingabedokumente werden standardmäßig in Preorder-Reihenfolge verarbeitet.
 - Bei Konflikten haben spezifischere Templates Vorrang vor weniger spezifischen.
 - Die Standard-Verarbeitungsreihenfolge kann durch ein Stylesheet geändert werden.
- (f) Wenn im XSL-Stylesheet keine Template-Regel für einen Knoten im Eingabedokument matched, geschieht folgendes:
- Die Abarbeitung des Eingabedokuments wird abgebrochen.
 - Der Knoten wird übersprungen.
 - Ein Built-in-Template wird angewandt.
 - Nichts davon.
- (g) Welche(r) der unten angegebenen Parser bilden die Struktur einer XML-Instanz-Datei im Speicher nach?
- SAX
 - DOM
 - JAXB
- (h) Welche(r) der unten angegebenen Parser ist Event-basiert?
- SAX
 - DOM
 - JAXB

Beachten Sie, dass zu einer Frage mehrere Antworten zutreffen können. Eine Frage gilt als richtig beantwortet, falls alle zutreffenden und keine unzutreffende Antwort angekreuzt ist.

Aufgabe 8 : Server Technologies: Multiple Choice

Kreuzen Sie Zutreffendes an:

(a) Folgende Programme und Dokumenttypen können unmittelbar als CGI-Programm verwendet werden:

- Jede Java-Servlet-Klasse
- Jedes in einer Shell ausführbare Programm
- Jedes Shell-Skript
- Jedes JSP-Dokument

(b) Wenn ein Java HTTP-Servlet durch eine HTTP-Anfrage angesprochen wird, führt der Tomcat-Servlet-Container jedes mal u. a. folgende Aktionen aus:

- Überprüfung ob eine Instanz dieses Servlets läuft
- Übersetzung des Servlets in den Maschinencode der Plattform des Browsers, von dem die Anfrage stammt
- Erzeugung eines `HttpServletRequest`-Objekts
- Starten der Java-Virtual-Machine

(c) Java `HTTPServlets` kommunizieren

- auf ISO/OSI-Schicht 1
- auf ISO/OSI-Schicht 2
- auf ISO/OSI-Schicht 3
- auf ISO/OSI-Schicht 4
- auf ISO/OSI-Schicht 5
- auf ISO/OSI-Schicht 6
- auf ISO/OSI-Schicht 7

(d) Java `Servlets` kommunizieren im Allgemeinen

- auf ISO/OSI-Schicht 1
- auf ISO/OSI-Schicht 2
- auf ISO/OSI-Schicht 2
- auf ISO/OSI-Schicht 3
- auf ISO/OSI-Schicht 4
- auf ISO/OSI-Schicht 5
- auf ISO/OSI-Schicht 6
- auf ISO/OSI-Schicht 7

(e) Welcher der aufgeführten regulären Ausdrücke erzeugt eine Sprache, die das Wort `rabimmelrabumm` enthält?

- `(rabimmelrabumm)*`
- `.*bimmel$`
- `(a|b|e|f|i|l|m)*`
- `(ra(bimmel|bumm))*`

Beachten Sie, dass zu einer Frage mehrere Antworten zutreffen können. Eine Frage gilt als richtig beantwortet, falls alle zutreffenden und keine unzutreffende Antwort angekreuzt ist.

Aufgabe 9 : DOM Manipulation

Laden Sie Sich die Datei `dom-manipulation.html` von der Kursseite herunter. Erstellen Sie die zugehörige `dom-manipulation.js` und implementieren Sie darin die unten geforderten Funktionalitäten. Ändern Sie die HTML-Datei *nicht*!

- (a) Fügen Sie einen Klick-Ereignishandler allen Listenelementen (``) hinzu, der das geklickte Element durch das Hinzufügen der Klasse "selected" auswählt. Hat das geklickte Element bereits die Klasse, so soll diese entfernt werden.
- (b) Fügen Sie einen Klick-Ereignishandler dem "Delete"-Knopf hinzu, der beim Klick alle ausgewählten Elemente entfernt.
- (c) Fügen Sie einen Klick-Ereignishandler den "Make Read" und "Make to Read"-Knöpfen hinzu, der beim Klick alle ausgewählten Elemente aus der aktuellen Liste entfernt und der jeweiligen Liste hinzufügt.