Nachklausur Elektronisches Publizieren, Sommer 2017

Name

Matrikelnummer

Sitzplatz

Sie können maximal 56 Punkte erreichen. Mit 22 Punkten bekommen Sie garantiert eine 4.0, mit 54 Punkten garantiert eine 1.0.

	1	2	3	4	5	6	7	8	9	10	11	12	13	∑
K														
NK														

Aufgabe 1 (Querschnitt) 7 Punkte

Erstellen Sie einen zusammenhängenden Text über den Vorlesungsinhalt, in dem die folgenden Begriffe vorkommen:

Element, XML DTD, XML-Technologie, Elementdeklaration, Typdefinition, XSLT, Document Management, CSS-Stylesheet, Garden of Eden, Venetian Blind, Russian Doll, Salami Slice, eXtensible Markup Language, XPath, XML Schema, Elektronisches Publizieren, Attribut, Document Engineering, Push-Schema, Pull-Schema, Relax NG, XQuery, Attributdeklaration, Inhaltstext, Knowledge Mangagement, global, lokal, FLOWR-Ausdruck

Unterstreichen Sie IN IHREM TEXT, wo die Begriffe aus der obigen Liste vorkommen. NUR UNTERSTRICHENE BEGRIFFE WERDEN BEWERTET.

Bewertung:
1/4 P für jeden korrekt beschriebenen und unterstrichenen Begriff, Endsumme auf halbe Punkte aufrunden.
28 Begriffe, max. 7 P

… noch Aufgabe 1 (Querschnitt)

Die drei Gebiete .Elektronisches Publizieren., .Document Management. und .Knowledge Management. bilden die Grundpfeiler des .Document Engineering.

XML steht für .eXtensible Markup Language. und ist durch das W3C standardisiert. Konzeptuell gesehen besteht ein XML-Dokument aus .Elementen., .Attributen. und .Inhaltstext.

Durch Schema-Sprachen wie .XML DTD., .XML Schema. und .Relax NG. kann definiert werden, wann XML-Dokumente valide sind. ...XML Schema... unterstützt vier Entwurfsmuster, nämlich ..Russian Doll.., ..Salami Slice.., ..Venetian Blind.. und ..Garden of Eden.. Diese Muster unterscheiden sich durch ihre .....globalen.... bzw. ....lokalen.... ..Elementdeklarationen.. und ..Typdefinitionen.. Das mächtigste der vier Muster ist ..Venetian Blind..

Erst der Einsatz von ..XML-Technologien.. wie ..XPath.., ..XSLT.. und ..XQuery.. macht XML operationabel. .....XSLT..... ist eine Programmiersprache, mit der XML-Dokumente in andere XML-Dokumente transformiert werden können. XSLT-Programme können mit dem ..Push-Shema... oder nach dem ..Pull-Schema.. organisiert. Mit .....XQuery..... können XML-Datenbestände abgefragt werden; das zentrale Sprachmittel ist dabei der ..FLOWR-Ausdruck…

Durch den Einsatz von ...CSS-Stylesheets.... können XML-Dokumente formatiert in Web-Browsern dargestellt werden.

Bewertung: 1 P für richtige Einordnung von (A), (B) und (C), und 1 P für richtige Bewertung der Attribute

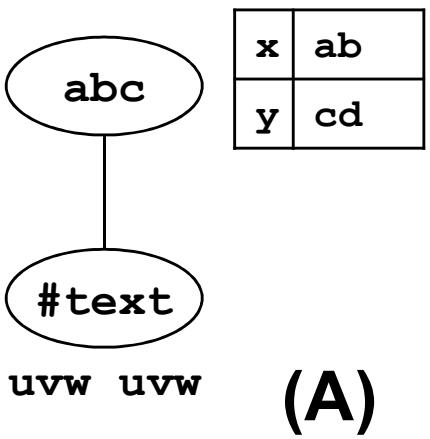
Aufgabe 2 (XML)

4 Punkte

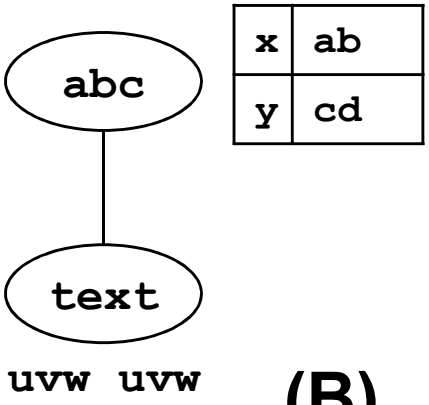
Betrachten Sie das folgende XML-Dokument:

```
<abc y="cd" x="ab">
  <text>uvw uvw</text>
</abc>
```

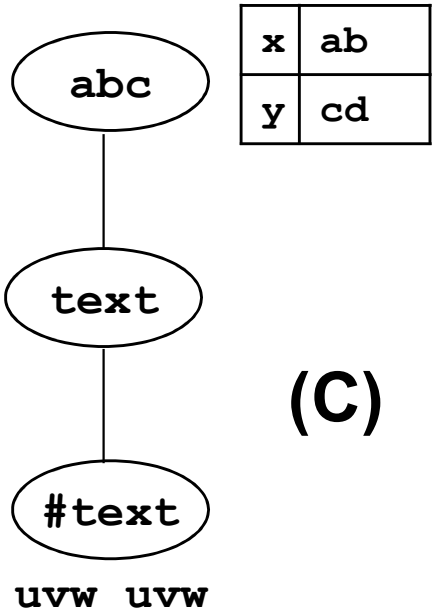
Welche der drei Baumdiagramme **(A)**, **(B)** oder **(C)** passen oder passen nicht zu dem Dokument, nach der in der Vorlesung präsentierten graphischen Notation? Begründen Sie jeweils Ihre Antwort.



(A)



(B)



(C)

… noch Aufgabe 2 (XML)

Das Dokument enthält ein Element mit Elementnamen Text, das einen Inhaltstext enthält. Im Diagramm muss also ein Elementknoten und ein Textknoten (mit Label #text) enthält. Nach diesem Kriterien sind die Diagramme (A) und (B) falsch und (C) ist korrekt.

Die Reihenfolge der Attribute ist in XML-Dokumenten irrelevant. Die Attributtabellen sind also in allen drei Diagrammen korrekt.

Damit ist also Diagramm (C) ein korrektes Abbild des Dokuments und Diagramme (A) und (B) sind nicht korrekt.

Aufgabe 3 (Namensräume) 4 Punkte

Betrachten Sie das nachfolgende Dokument:

```
<s:x version="0" xmlns:s="http://www.bsp.de/a">
    <x version="1">
        <x version="2" xmlns="http://www.bsp.de/a">
            <x version="3" xmlns=""/>
        </x>
        <s:x version="4" xmlns:t="http://www.bsp.de/b">
            <t:x version="5" xmlns:t="http://www.bsp.de/a"/>
            <t:x version="6">
                <x version="7"/>
            </t:x>
        </s:x>
    </x>
    <t:x version="8"/>
</s:x>
```

… noch Aufgabe 3 (Namensräume)

Ordnen Sie in der nachfolgenden Tabelle den Namensraum-URIs die Elemente mit ihren Versionsnummern zu, die zu dem jeweiligen Namensraum gehören. Beachten Sie, dass für manche Elemente keine Namensraumzuordnung gegeben ist (Fehler aus Sicht von XML Namespaces).

Namensraum-URI	Wert von Attribut version in Element x `<x version="…">…</x>`
`http://www.bsp.de/a`	0, **2, 4, 5**
`http://www.bsp.de/b`	**6**
Anonymer Namensraum ("")	**1, 3, 7**
Fehler	**8**

Bewertung:

1/2 P pro richtiger Zuordnung.

8 Zuordnungen, max. 4 Punkte.

Aufgabe 4 (Datentypen) 1.5+1.5+1+1 Punkte

- Was ist ein Datentyp in einer Schemasprache für XML und wie kann er verwendet werden? Beschreiben Sie drei deutlich voneinander verschiedene Aspekte von Datentypen. (1) Die Werte eines Datentyps sind Strings von Unicode-Zeichen. (2) Je nach Datentyp folgen die String-Werte gewissen Bildungsmustern (z.B. 31.8E2 oder 2017-01-02). (3) Datentypen können als Typen für Attribute und reinen Textinhalt von Elementen verwendet werden. (4) Die Werte eines Datentyps können als Attributwerte oder als Inhalt eines Elements verwendet werden. (5) Es gibt je nach Schemasprache einen unterschiedlichen Satz an vordefinierten Datentypen.
- Nennen und beschreiben Sie den Mechanismus, mit dem Sie in XML Schema einen eigenen Datentypen definieren können, der den Wertebereich eines vorgegebenen Datentyps einschränkt. Verwenden Sie die Begriffe Facette und Pattern. Der Mechanismus heißt Restriction. Facetten sind die Kriterien, nach denen ein Basistyp eingeschränkt werden kann, z.B. minInclusive oder maxExclusive für numerische Datentypen. Patterns sind spezielle Facetten, die einen regulären Ausdruck angeben, nach dem die Werte des Basistyps eingeschränkt werden sollen.
- Wie heißen die anderen beiden Mechanismen, mit denen Sie in XML Schema einen eigenen Datentypen definieren können? Union (Vereinigung) und List (Listenbildung)
- Gibt es in XML-Schema vordefinierte Strukturtypen? NEIN. Vordefinierte Datentypen? JA.

Aufgabe 5 (DTD) 2 Punkte

Nehmen Sie an, Sie haben eine DTD, die das Dokument <x><x/><x/></x> validiert.

- Was ist das kleinste Dokument, das diese DTD ebenfalls validieren muss ?
- Geben Sie ein weiteres Dokument an, das diese DTD validiert.

Das Dokument hat ein Wurzelelement x mit zwei Kindelementen x. Eine DTD, die dieses Dokument validiert, erlaubt für ein Element x also den leeren Inhalt und alternativ genau zwei Unterelemente. Das kleinste Dokument, das die DTD akzeptiert, ist also <x/>. Ein weiteres zulässiges Dokument ist <x> <x><x/><x/></x> <x/> </x>.

Aufgabe 6 (XML Schema) 1+2+1 Punkte

Betrachten Sie das XML Schema auf der folgenden Seite.

- Nach welchem Entwurfsmuster ist dieses Schema aufgebaut? Begründen Sie Ihre Antwort. Venetian Blind, globale Typdefinitionen und lokale Elementdeklarationen, mit Ausnahme des Einstiegselements.
- Im allgemeinen können Sie Schemata, die nach diesem Entwurfsmuster aufgebaut sind, nicht äquivalent in das Salami-Slice-Muster umformen (äquivalent: beide Schemata validieren die gleichen Dokumente). Begründen Sie diese Aussage. In Venetian Blind gibt es lokale Elementdeklarationen, in Salami Slice nicht. In Venetian Blind können also zwei gleichnamige Elemente mit verschiedenen Typdefinitionen verbunden werden, während in Salami Slice ein Element mit Namen XXX immer mit der gleichen Typdefinition verbunden ist.
- In diesem konkreten Beispiel ist es jedoch möglich, das Schema äquivalent in das Salami-Slice-Muster umzuformen. Begründen Sie diese Aussage. In dem Beispiel ist jeder Elementname ist nur einfach deklariert; d.h. Elementdeklarationen können global gemacht werden. Globale Typdefinitionen können immer lokal gemacht werden. In diesem Fall, wo es keine Rekursionen gibt, ist es direkt offensichtlich.

… noch Aufgabe 6 (XML Schema)

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="book" type="bookType"/>
   <xs:complexType name="bookType">
      <xs:sequence>
         <xs:element name="author" type="xs:string"/>
         <xs:element name="character" type="characterType" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="characterType">
      <xs:sequence>
         <xs:element name="name" type="xs:string"/>
         <xs:element name="friend-of" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
         <xs:element name="since" type="xs:date"/>
         <xs:element name="qualification" type="xs:string"/>
      </xs:sequence>
   </xs:complexType>
</xs:schema>
```

Aufgabe 7 (XSLT) 2+2 Punkte

Betrachten Sie das folgende XSLT-Programm:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:transform xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:output method="xml" encoding="UTF-8" indent="yes"/>
    <xsl:template match="/"> <xsl:apply-templates/> </xsl:template>
    <xsl:template match="*">
        <xsl:copy> <xsl:apply-templates/> </xsl:copy>
    </xsl:template>
    <xsl:template match="z">
        <xsl:copy> <xsl:apply-templates select="@*"/> </xsl:copy>
    </xsl:template>
    <xsl:template match="@*"> <xsl:copy/> </xsl:template>
    <xsl:template match="abc"> <xsl:apply-templates select="xyz"/>
      </xsl:template>
    <xsl:template match="xyz"> <xsl:copy/> </xsl:template>
    <xsl:template match="text()">ZZZ</xsl:template>
</xsl:transform>
```

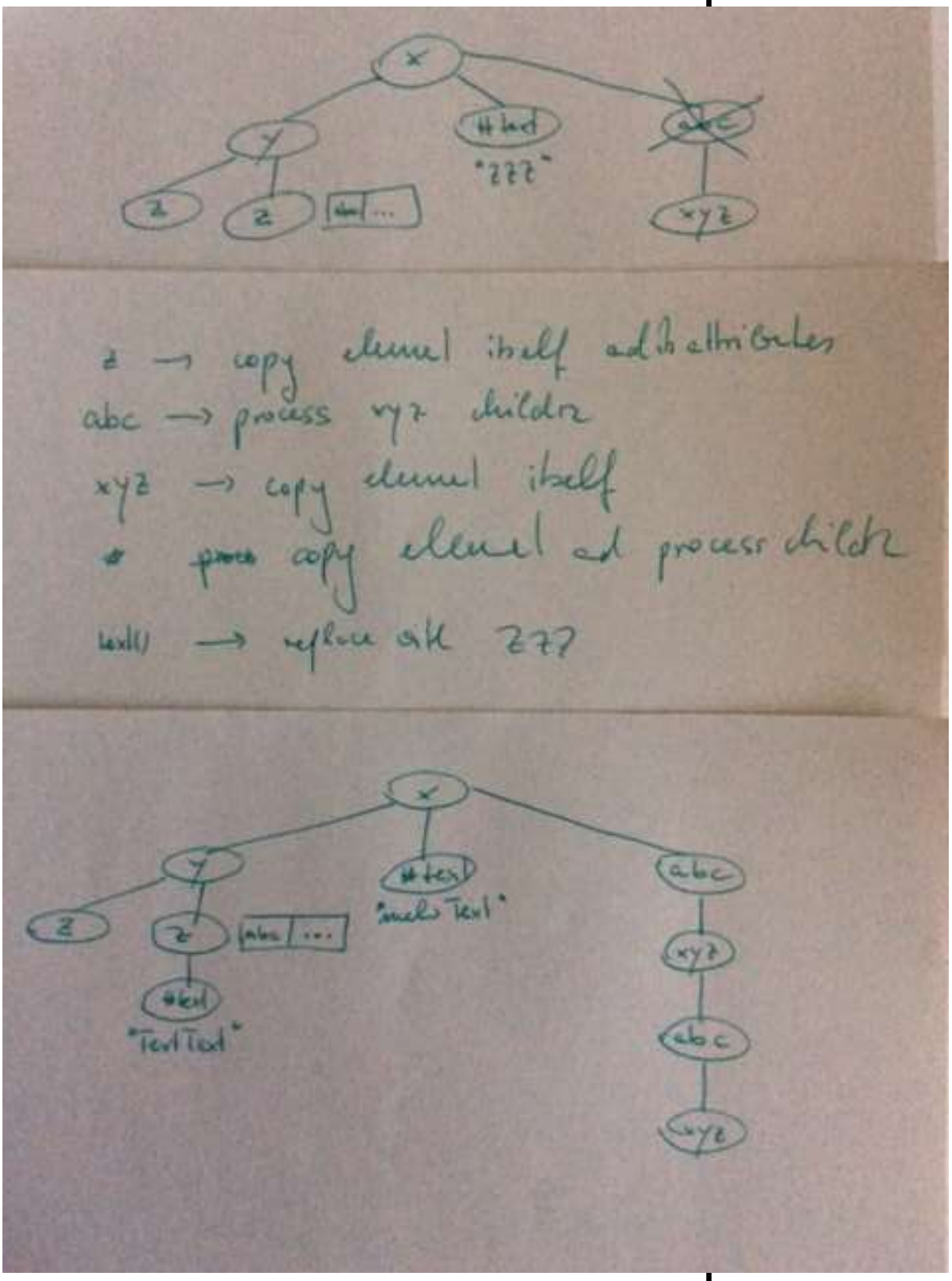
… noch Aufgabe 7 (XSLT)

Bewertung:
1/2 P Abzug für jeden Fehler,
max. 2 P für Baumdiagramm Input,
max. 2 P für Baumdiagramm/XML Output

Erstellen Sie ein Baumdiagramm für das folgende XML-Dokument und wenden Sie dann das XSLT-Programm von der vorigen Seite auf dieses Dokument an. Stellen Sie das Ergebnis als Baumdiagramm oder als XML-Dokument dar.

Hinweis: Ausgehend von einem Element als Kontextknoten, evaluiert * zu der Sequenz der Kinde*lemente* und @* zu der Sequenz der (kompletten) Attributknoten.

```
<x><y><z/><z abc="...">TextText</z></y>
    mehr Text
    <abc><xyz><abc><xyz/></abc></xyz></abc></x>
```


x
y
"777"
xyz
z → copy element itself and its attributes
abc → process xyz children
xyz → copy element itself
* copy element and process children
text() → replace with 777
x
y
"multi Text"
abc
xyz
abc
xyz
"Text Text"

Bewertung: 16 Begriffe, 1/4 Punkt für jeden richtig verwendeten und im Text unterstrichenen Begriff. Endsumme auf halbe Punkte aufrunden.

Aufgabe 8 (XPath, XQuery) 4 Punkte

Erstellen Sie einen zusammenhängenden Text über XPath und XQuery, in dem die folgenden Begriffe vorkommen: Prädikat, Achse, Kind-Achse, Slash (/), Knotentest, Step, Kontextknoten, FLWOR-Ausdruck, Abfragesprache, Teilsprache, ist berechenbarkeits-universell, ist nicht berechenbarkeits-universell, XDM, SQL, relationales Modell, Datenmodell

Unterstreichen Sie IN IHREM TEXT, wo die Begriffe aus der obigen Liste vorkommen. NUR UNTERSTRICHENE BEGRIFFE WERDEN BEWERTET.

Ein Location-Path Ausdruck in XPath besteht aus einer Folge von Steps, die durch einen Slash (/) voneinander getrennt sind. Ein Step besteht aus einer Achse, einem Knotentest und optionalen Prädikaten. Die Default-Achse ist die Kind-Achse.

Ein XPath-Ausdruck wird von einem Kontextknoten ausgehend ausgewertet.

XPath ist eine Teilsprache von XQuery. Das zentrale Sprachmittel von XQuery ist der FLWOR-Ausdruck. XQuery ist berechenbarkeits-universell, XPath ist nicht berechenbarkeits-universell.

XPath und XQuery basieren auf dem Datenmodell XDM. Es bestehen Parallelen zwischen der XML-Welt und der Welt der relationalen Daten: XDM entspricht dem relationalen Modell und XQuery entspricht SQL.

Bewertung: Jeweils 1 P für korrekte Ausgabe und 1 P für korrekte Begründung.

Aufgabe 9 (XPath) 4 Punkte

Betrachten Sie das folgende XML-Dokument:

```
<x>
   <p y="abc" no="1"/>
   <p y="abc" no="2"/>
   <p y="uvw" no="3"/>
   <p y="abc" no="4"/>
   <p y="abc" no="5"/>
</x>
```

1. Ausdruck: Es werden mit dem ersten Prädikat alle Elemente namens p selektiert, die ein Attribut y mit dem Wert "abc" haben, also die Elemente mit Nummern 1, 2, 4 und 5. Von denen wird in dem zweiten Prädikat das vierte Element selektiert, also <p y="abc" no="5"/>.

Welche Ausgabe ergibt sich jeweils, wenn Sie die folgenden beiden XPath-Ausdrücke gegenüber dem Dokument evaluieren? Beachten Sie, dass ein Filterausdruck [k] für eine ganze Zahl k von der aktuell betrachteten Sequenz für das Kontextelement an Position k zu true evaluiert und dass die Numerierung von Positionen bei 1 beginnt. Begründen Sie Ihre Antworten.

- **//p[@y='abc'][4]**
- **//p[4][@y='abc']**

2. Ausdruck: Es wird mit dem ersten Prädikat das Element p an Position 4 selektiert, also der Knoten <p y="abc" no="4"/>. Das zweite Prädikat evaluiert für diesen Knoten zu "true", die Ausgabe ist also also dieser Knoten mit Nr 4.

Bewertung: ½ P Abzug für jede fehlende/falsche Angabe. Einmalig ½ Punkt Abzug für fehlende Klammern <number>…</number>.

Aufgabe 10 (XQuery) 4 Punkte

Welche Ausgabe liefern jeweils die folgenden XQuery-Ausdrücke? Genau hinschauen, Änderungen gegenüber der Probeklausur!!!

```
let $i:=(2,4,6)
return <number>{2*($i[.>5])}</number>
```

Ausgabe:

```
<number>12</number>
```

```
for $i in (1,2,3)
return <number>$i+3</number>
```

Ausgabe:

```
<number>$i+3</number>
<number>$i+3</number>
<number>$i+3</number>
```

```
for $i in (2,2)
for $k in (1,2,4)
return <number>{$i*$k}</number>
```

Ausgabe:

```
<number>2</number>
<number>4</number>
<number>8</number>
<number>2</number>
<number>4</number>
<number>8</number>
```

Aufgabe 11 (XQuery) 2 Punke

Geben Sie das Ergebnis der folgenden XQuery an. Beachten Sie dabei, dass der Ausdruck "k to l" für ganze Zahlen k und l zur Sequenz (k,…,l) evaluiert. "1 to 3" steht also für die Sequenz (1,2,3). Eine ganze Zahl k ist äquivalent zu der Sequenz (k) mit nur einem Item.

```
for $a in 3 to 4
for $b in 1 to 3
where $a * $b > 5
return for $c in 4
       let $d :=($a, $b, $c)
       return <X>{ $d }</X>
```

Ausgabe:

Bewertung:

1 P pro korrekter Ausgabe

1 P Abzug pro falscher Ausgabe

0.5 P Abzug global für fehlende <X>…</X> Umklammerung

Aufgabe 12 (Unicode) 2+5 Punkte

- Eine Datei besteht aus 30 **Zeichen** aus dem Zeichensatz ISO-Latin-1. Sie sollen die Datei in UTF-8 kodieren. Aus wievielen **Bytes** wird die UTF-8-kodierte Datei maximal bestehen. Geben Sie eine dichte obere Schranke an, die tatsächlich erreicht werden kann. Begründen Sie Ihre Antwort.
  Die ersten 256 Zeichen in der Unicode-Zeichentabelle sind genau die Zeichen aus ISO-Latin-1. UTF-8 als Zeichenkodierung variabler Länge kodiert die ersten 128 Zeichen der Unicode-Zeichentabelle mit einem Byte und die weiteren 128 (…und mehr…) Zeichen mit 2 Bytes. Falls die Datei also nur Zeichen aus der unteren Hälfte von ISO-Latin-1 enthält, wird unter UTF-8 jedes dieser Zeichen mit zwei Bytes kodiert und die Datei enthält genau 60 Bytes. Das ist eine obere Grenze.

- Setzen Sie die folgenden Textstücke in den Lückentext ein, der auf der rechten Seite steht. Passen Sie die grammatischen Formen geeignet an. Die Textstücke können mehrfach verwendet werden. Es müssen nicht alle Textstücke verwendet werden.

  Textstücke: Basic Multilingual Plane, Ziffer, Adressraum, Kachel, arabisch, Zeichenstrom, ISO-Latin I, Speichersicht, lateinisches Alphabet, Zeichensatz, 21 Bit, US-ASCII, Sonderzeichen / Umlaute, Positionen, Zeichentabelle, Kodierungsformat, europäisch, 16 Bit, Ebene, XML-Sicht, weltweit, Ziffer, UCS, UTF-8, 8 Bit, Gruppe

… noch Aufgabe 12 (Unicode)

Der Unicode-Zeichensatz .....UCS..... ist in einer abstrakten ..Zeichentabelle.. angeordnet, die in 17 .....Ebenen..... unterteilt ist. Die erste ....Ebene..... heißt ..Basic Multilingual Plane.. Jede .....Ebene..... besteht aus 256 .....Kacheln..... mit je 256 .....Positionen..... Ein .....Kodierungsformat... (z.B. ....UTF-8.....) legt fest, wie die Positionen für die einzelnen Zeichen in Bits oder Bytes abgebildet werden. Aus .....Speichersicht.. besteht ein Zeichenstrom aus einer Sequenz von Bytes. Aus …….....XML-Sicht…….. besteht ein Zeichenstrom aus einer Sequenz von Unicode-Zeichenpositionen. ....Kachel..... 0 in .....Ebene..... 0 ist identisch mit .....ISO-Latin-1...., was wiederum in zwei Hälften aufgeteilt ist. Die obere Hälfte, ...US-ASCII....., enthält Zeichen aus dem .lateinischen Alphabet., jedoch keine ...Sonderzeichen / Umlaute... Das ..Kodierungsformat.. namens .....UTF-8..... ist US-ASCII-transparent.

Bewertung der zweiten Teilaufgabe:
1/4 P für jeden korrekt eingefügten Begriff,
Endsumme auf halbe Punkte aufrunden. 20
Lücken, max. 5 P

Aufgabe 13 (XSL-FO) 2 + 2 +1 Punkte

- Ein XSL-FO-Prozessor übersetzt ein XSL-FO-Dokument in ein Präsentationsformat (z.B. PDF). Der XSL-FO-Prozessor arbeitet dabei in zwei Phasen und erzeugt dabei ein Zwischenformat.

  Wie heißt das Zwischenformat und was für Information enthält es?
  Das Zwischenformat heißt Area Tree. Es enthält das Ergebnis der Formatierung in abstrakter Form, also eine genaue Beschreibung der Texte mit Fonts und Positionen auf den Seiten.

- Wie heißen die beiden Phasen und was haben Sie für Aufgaben?
  Die Phasen heißen Formatierung und Rendering. Die Formatierung bestimmt Zeilen- und Seitenumbrüche, typographische Attribute für Schriften, Abstände etc. und hat den Area Tree als Ausgabe. Das Rendering zeigt den Area Tree übersetzt den Area Tree in ein spezifisches Ausgabeformat, z.B. nach PDF, RTF oder GIF.

- Welchen Vorteil bietet die Aufteilung in die beiden Phasen mit standardisiertem Zwischenformat?
  Die aufwendige Aufgabe des Formatierens kann unabhängig vom Ausgabeformat realisiert werden und ist für die verschiedenen Ausgabeformate wiederverwendbar. Das gilt für die Entwicklung eines XSL-FO-Prozessors als Software und auch für die Formatierung einzelner Dokumente.

Leere Seite

Leere Seite