

**ACHTUNG! Für dieses Übungsblatt kann bis
SPÄTESTENS DIESTAG 29.11.2005 UM 17 UHR
gekennzeichnet werden!**

Aufgabe 1

Gegeben ist folgender Ausschnitt aus einer XML-Datei:

```
<homepage name="Datenbanksysteme">
  <server>www.cs.univie.ac.at</server>
  <path>/dbs</path>
  <port>80</port>
</homepage>
```

Schreiben Sie für den XML-Ausschnitt eine passende **DTD** unter folgender Zusatzannahmen: Das Rotelement heisst `homepages`, es enthält ein oder mehrere `homepage` Elemente. Falls Sie weitere Zusatzannahmen treffen, dokumentieren Sie diese schriftlich.

Aufgabe 2

Gegeben ist der XML-Ausschnitt aus Aufgabe 1. Lösen Sie folgende Abfragen mittels **XPath**:

- a) Die Namen aller Homepages, die am Server `www.cs.univie.ac.at` gehostet werden.
- b) Die serverseitigen Pfade (`path`) aller Homepages, die nicht auf Port 80 gehostet werden.
- c) Server und Pfad der oben abgebildeten "Datenbanksysteme" Homepage.
- d) Den Namen der letzten in der Datei gespeicherten Homepage mit leerem Pfad.

Aufgabe 3

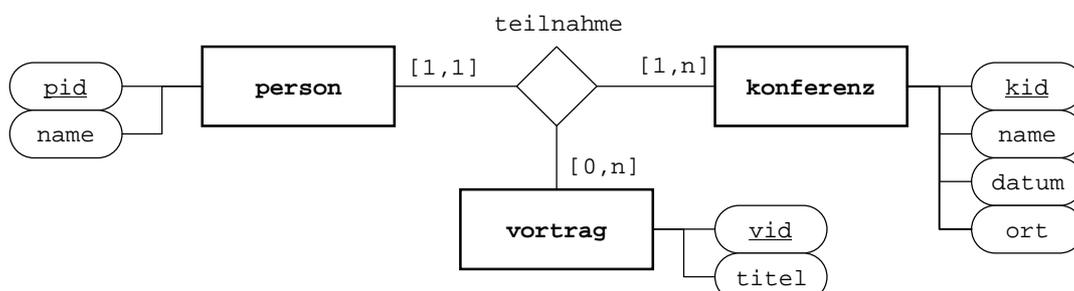
Gegeben ist der XML-Ausschnitt aus Aufgabe 1. Schreiben Sie eine **XQuery**, die alle Homepages in der XML-Datei als sortierte Liste von HTML-Hyperlinks (URLs) ausgibt, und zwar sortiert nach den Namen der Homepages. Für die Beispielhomepage im Ausschnitt müsste das Ergebnis folgendermaßen aussehen:

```
<a href="http://www.cs.univie.ac.at:80/dbs">Datenbanksysteme</a>
      á          á á          á
      server    port path    name
```

(Hinweis: Sie können XQuery Ausdrücke unter <http://elearn.pri.univie.ac.at/tools/xquery> ausprobieren. Dieses Webtool basiert auf AltovaXML 2006, das Sie sich auch lokal als Kommandozeilentool herunterladen können unter <http://www.altova.com/altovaxml.html>)

Aufgabe 4

Sie entwickeln gerade eine Webapplikation für eine Konferenzverwaltung. Das ER-Diagramm Ihres zu Grunde liegenden Datenmodells ist unten abgebildet. Sie verwenden XML zur Datenhaltung.



- Schreiben Sie eine DTD, welche die im Klassendiagramm modellierten Datenstrukturen und deren Beziehungen strukturell korrekt abbildet (*Hinweis*: Es ist am einfachsten, wenn Sie als ersten Schritt aus dem ER-Diagramm ein relationales Datenbankschema ableiten, und daraus dann die DTD.)
- Schreiben Sie einen XPath-Ausdruck, der die Namen aller Personen liefert, die jemals auf Konferenzen in Wien einen Vortrag zum Thema "Datenbanksysteme" gehalten haben.

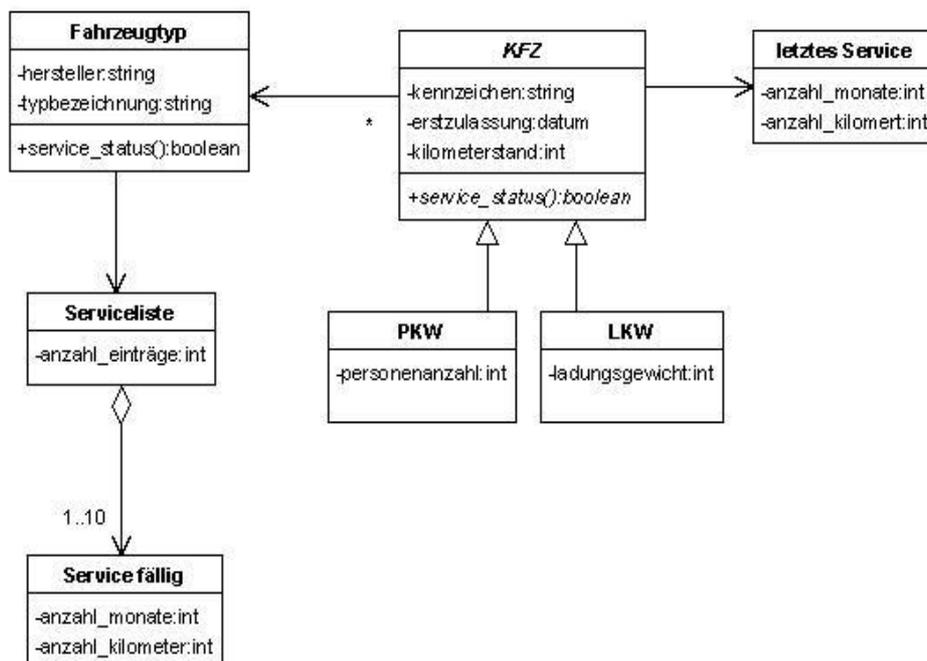
Aufgabe 5

Gegeben ist folgende DTD. Schreiben Sie eine für diese DTD gültige **XML Datei**.

```
<!ELEMENT fehlerdatenbank (fehler, fehler+)>
<!ELEMENT fehler (finder, beschreibung, kategorie, kommentar*)>
<!ELEMENT finder (#PCDATA)>
<!ELEMENT beschreibung (#PCDATA)>
<!ELEMENT kategorie (#PCDATA)>
<!ELEMENT kommentar (#PCDATA)>
<!ATTLIST fehler
  id NMTOKEN #REQUIRED
  funddatum CDATA #REQUIRED
  behoben (ja|nein) "nein">
<!ATTLIST kategorie ernsthaftigkeit (naja|pfui|auweh) #REQUIRED>
<!ATTLIST kommentar autor CDATA #REQUIRED>
```

Aufgabe 6

(Für die Aufgaben 6 bis 10 gilt allgemein: Versuchen Sie Unklarheiten in den Spezifikationen durch geeignete Zusatzannahmen zu lösen.)



Zur Verwaltung eines Fuhrparks soll eine objektrelationale Datenbank verwendet werden, die das obige UML Diagramm abbildet (Kardinalitäten, die bei den Assoziationen nicht extra angeführt werden, sind als 1 anzunehmen). Für jedes KFZ werden das Alter des Fahrzeuges (*anzahl_monate*) und der Kilometerstand (*anzahl_kilometer*) beim letzten durchgeführten Service gespeichert. Für jeden Fahrzeugtyp gibt es eine Liste des Herstellers, bei welchem Alter bzw. Kilometerstand die Services durchzuführen sind.

Geben Sie geeignete Typ- und Tabellendefinitionen zum Speichern der Informationen an. (Die diversen `service_status` Methoden brauchen Sie nicht zu implementieren.)

Aufgabe 7

Nehmen Sie an, dass die Funktion `service_status` in der Klasse `KFZ` `true` retourniert, falls das Fahrzeug zum Service sollte, `false` sonst. Schreiben Sie eine SQL-Query, die alle Fahrzeuge listet, für die ein Service ansteht. Für jedes dieser Fahrzeuge soll auch eine Liste der nächsten Servicetermine laut Hersteller ausgegeben werden.

Aufgabe 8

Implementieren Sie die Methoden `service_status` für die Klassen `PKW`, `LKW` und `Fahrzeugtyp` so, dass der Aufruf der Methode für ein bestimmtes Fahrzeug liefert, ob ein Servicetermin ansteht oder nicht. Die Methoden sollen berücksichtigen, dass zusätzlich zu den, vom Hersteller vorgegebenen Terminen `PKWs` alle Jahre und `LKWs` alle 5 Monate zum Service müssen.

Aufgabe 9

Oracle unterstützt keine Tabellenvererbung. Um zu gewährleisten, dass alle `PKW` Objekte auch in der Tabelle `KFZ` repräsentiert sind, können Trigger verwendet werden. Schreiben Sie Trigger für `Insert`-, `Update`- und `Delete`-Statements auf der Tabelle `PKW`, die die Daten in der Tabelle `KFZ` entsprechend nachführen. Schreiben Sie auch die Trigger für `Update`- und `Delete`-Statements auf der Tabelle `KFZ`, die die Daten in der Tabelle `PKW` nachführen.

Aufgabe 10

Die Fahrzeuge des Fuhrparks werden auf einem Parkplatz abgestellt, der ebenfalls in einer Tabelle verwaltet werden soll. In dieser Tabelle soll jeweils die Nummer des Parkplatzes und eine Objektreferenz auf das geparkte Fahrzeug gespeichert werden, bzw. `NULL`, falls der Parkplatz leer ist. Geben Sie die Definition der Tabelle, sowie eventuell notwendige Änderungen an den bereits vorher festgelegten Tabellendefinitionen an und erstellen Sie eine Query, die die Nummern aller Parkplätze ermittelt, auf denen Fahrzeuge stehen, die zum Service müssten.

Anmerkungen

Da die Syntax vor allem von `ORACLE` doch zum Teil erheblich von den Beispielen im Skriptum abweicht, werden hier ein paar Beispiele aus dem Skriptum noch einmal `ORACLE-like` aufgearbeitet:

```
create type employee as object(name varchar(20), salary integer) not final;
```

Beachte Verwendung des Schlüsselworts 'object'. 'not final' muss verwendet werden, wenn von dem Typ weitere Datentypen abgeleitet werden sollen.

Arrays müssen vorher als Typ definiert werden, sie können nicht direkt in einer Tabellendefinition definiert werden.

Methoden:

Deklaration mit

```
create type employee as object(name varchar(20), salary integer, member function giverraise return int);
```

Definition:

```
create type body employee as
  member function giverraise return int is
  x int -- lokale variable
  begin
    -- funktionsbody
  end;
end;
/
```

Überladen einer methode mit dem keyword 'overriding' vor dem keyword 'member' (und zwar sowohl bei der Deklaration, als auch bei der Definition.)

Referenzen:

```
create type department(name varchar(20), head ref person);
```

Ein Tip zu XQuery noch:

Sollen im return Teil die Werte von Variablen ausgegeben werden, dann müssen geschwungene Klammern verwendet werden.

z.B.:

```
for ...return <element>{$x}</element>
```

Zur Ausgabe von Attributwerten können sie die Funktion string() verwenden.