

---

# PRÜFUNGSprotokoll EBT & MMDB

DER PRÜFLING  
February 17, 2016

---

## Contents

1	eBusiness Technologies	1
2	Multimedia-Datenbanken	2

---

**Prüfer** Prof. Dr. Klaus Meyer-Wegener, unbekannter Prüfungsprotokollführer

**Version** 1.0 of 2016-02-17

**Stimmung** Gelassen

**Veranstaltungsbesuch** Weder in MMDB noch in EBT.

**Vorbereitung in Tagen** 3 Tage

**Vorbereitungsstil** Ganz normal gelernt. 3 Tage sind aber definitiv zu wenig. 5 Tage wären vermutlich optimal gewesen.

## 1 eBusiness Technologies

1. Beschreiben sie die **Bedeutung** und das **Ziel** von
  - JSF (Code in HTML, HTML in Code erklären)
  - RUMPS (*oder so...*)
  - UDDI (Webservice discovery)
  - Rational Unified Process (Rati)
  - CGI (stdin, stdout, echo/print, pro Request ein Prozess -> langsam)
  - AJAX (Javascript, CSS, DOM, XMLHttpRequest)

- DAO
  - EJB (Entity Bean, Session Bean, Message Bean)
  - MVC (am besten aufmalen)
2. Welche Komponenten stellt ein Application Server der Anwendung zur Verfügung?

Hauptsächlich sog. Cross-Cutting-Concerns wie:

- Logging
  - Datenbankanbindung
  - Transaktionen
  - Messaging
  - Security
3. Was ist denn O/R mapping?

Dazu hab ich den Impedence Mismatch erklärt (Objekte <-> Relationen). Das geht in zwei Richtungen:

Quelle	Ziel	Beispiel
Klassen in Java	Datenbankschema	Hibernate
Datenbankschema	Klassen in Java	jOOQ

4. Wie kann man Hierarchien in relationale Datenbanken darstellen.

Das sind im Prinzip die drei Methoden von Elmasri-Navathe aus KonzMod. Angenommen man hat den Typ Mensch und Dozent und Student sind Subtypen davon:

- Alles in eine Tabelle (aber: viele null-Werte bei den Unterschieden)
  - Jeweils eine eigene Tabelle für jeden Typ (aber: Name ist z.B. doppelt gespeichert)
  - Eine gemeinsame Tabelle Mensch und dann mit 1:1 zu Dozent und Student (aber: fast immer ein JOIN nötig)
5. (Wurde nicht gefragt, habs aber auf dem Zettel gesehen):  
Was ist AOP? Wo und warum setzt man das ein?

## 2 Multimedia-Datenbanken

1. Wir bauen eine Fotoverwaltung. Erstellen sie dazu mal ein Schema:

```
CREATE TABLE Foto (
  foto_id integer PRIMARY KEY,
  foto Image
  --    `-> User defined type
)
```

2. Was gehört doch neben diesem Typ noch dazu?

Hier wollte er „Verhalten“ hören, also Funktionen die auf dem Typ „Image“ arbeiten.

3. Was gibt es denn da so für Funktionen für „Image“?

- getWidth()
- getHeight()
- getObjects()
- getMedianColor()
- replacePart(x0, y0, x1, y1, Image)
- setPixelRGB(x, y, r, g, b)

Allgemein also lesende- und schreibende Anfragen.

4. Wir fügen jetzt mal ein Foto in die Datenbank ein. Wie könnte das aussehen?

```
-- Aus einem Programm heraus
INSERT INTO Foto VALUES (0, 0, Image(:fotobytes))
--                               ^-> Konstruktor

-- In einer interaktiven Umgebung
INSERT INTO Foto VALUES (0, 0, Image("IMG_001.jpg"))
--                               ^-> Konstruktor
```

Es ging ihm hier speziell um den Konstruktor, definiert in den Funktionen aus 2.

5. Jetzt lesen wir mal ein Foto. Wie sieht das in SQL aus?

```
SELECT foto.asJPG() from Foto where foto_id=0
```

Hier war es wichtig zu sagen dass wir das Foto konvertieren müssen, weil das interne Format vermutlich kein JPG sein wird. Datenunabhängigkeit.

6. Wir würden jetzt gerne zwei Fotos anhand des Histogramms vergleichen. Schreiben Sie dazu bitte das SQL hin.

```
SELECT foto_id FROM Foto where foto.compare_histogram(Image("IMG_002.jpg")) > 0.7
```

Hier war es wichtig zu erwähnen, dass ``compare\_histogram`` eine Funktion für den User Definiert „Image“ ist.

7. Aus was bestehen denn grundsätzlich Multimediaelemente in einer Datenbank?

- Rohdaten (z.B. Bytes)
- Registrierungsdaten (Daten zur Interpretation der Rohdaten)
- Beschreibungsdaten (optional, Beschreiben den Inhalt meist in einem anderen Medium)

8. Was kann denn in solchen Beschreibungsdaten z.B. bei Fotos alles drinstehen?

- Schlagwörter
- Histogramme
- Farben des Bildes (meist irgendwie aggregiert, z.B. als Mittelwert)
- Auf dem Bild sichtbare Objekte
- Formen

- Texturen

Danach kann man dann natürlich auch suchen.

9. Man könnte ja jetzt das Bild auch als BLOB speichern. Warum machen wir das nicht?

Datenunabhängigkeit. Man möchte eine interne Restrukturierung machen können ohne dass sich auch nur ein einziger Query verändert. Außerdem hätte man gerne die Möglichkeit verschiedene Formate (`foto.asJpg()` / `foto.asPng()`) herauszugeben.

---