



Web- Entwicklung II

Wintersemester 2017/18

Übungen zum Kapitel 4

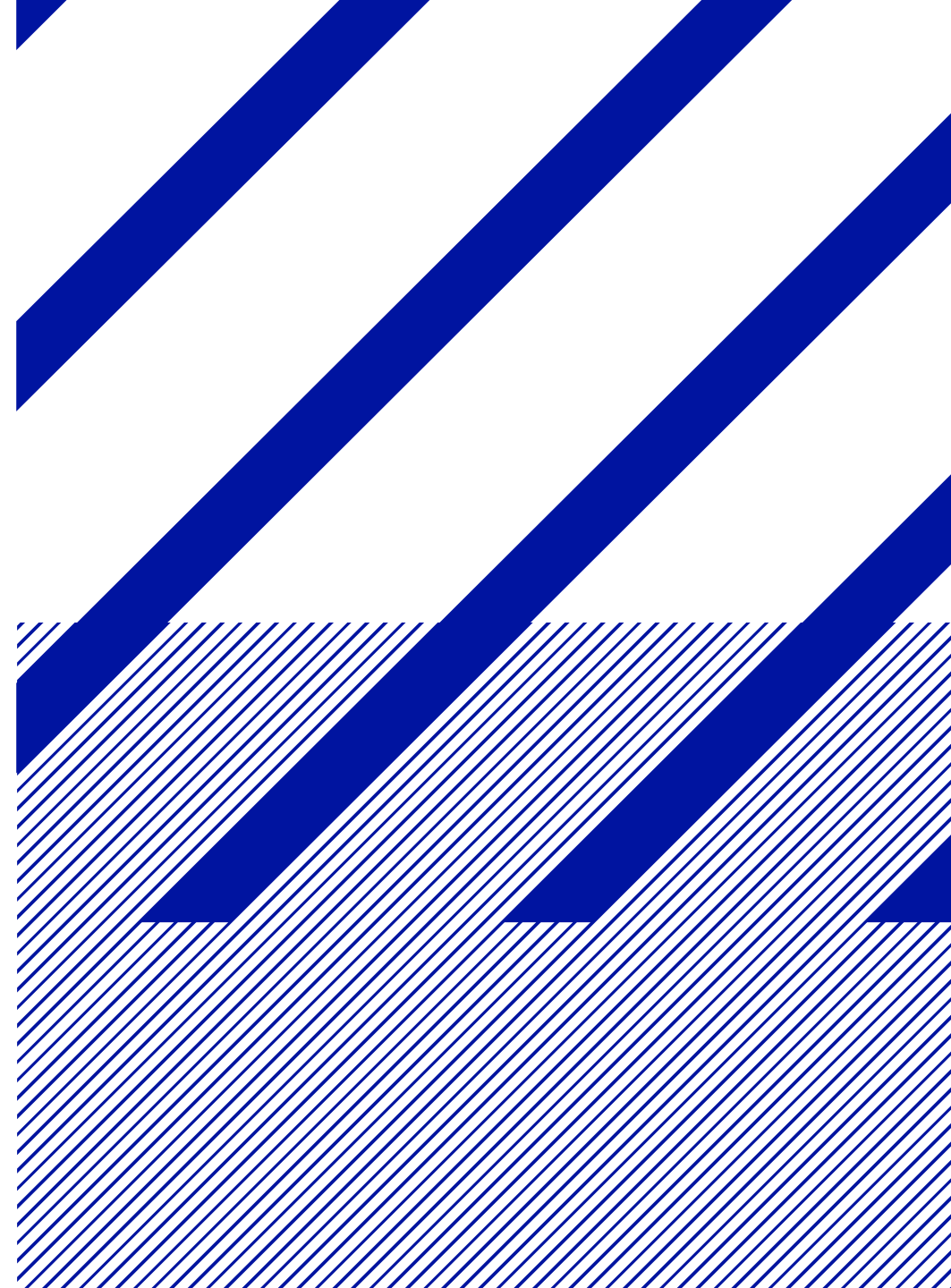
Prof. Dr. Norman Lahme-Hütig

Informatik/Wirtschaftsinformatik, Schwerpunkt Web Engineering

Corrensstraße 25
D-48149 Münster

fon +49 (0)251.83 65-190
fax +49 (0)251.83 65-525

norman.lahme-huetig@fh-muenster.de
www.fh-muenster.de



Übung 1

Aufgaben: TypeScript-Projekt



1. Ihr express-tm-Projekt soll nun schrittweise in ein TypeScript-Projekt überführt werden. Beginnen Sie wie folgt:
 - Erstellen Sie die Konfigurationsdatei `tsconfig.json`
 - Ändern Sie bei den Dateien `app.js` und `tasks.js` die Dateinamen zu `app.ts` und `tasks.ts`
 - Transpilieren Sie die Anwendung. Was fällt Ihnen auf?
2. Ändern Sie den Konfigurationsparameter `strict` zu `false` und transpilieren Sie die Anwendung erneut. Was stellen Sie fest?
3. Installieren Sie nun Typings-Dateien für Node.js sowie für alle verwendeten Pakete und transpilieren Sie anschließend die Anwendung erneut. Gibt es weiterhin Fehler? Falls ja, warum?

```
express-tm
  db
  node_modules
  src
    public
    routes
    tasks.ts
  views
  app.ts
  package.json
  tsconfig.json
```

Übung 1

Aufgaben: TypeScript-Projekt



4. Ändern Sie in den beiden .ts-Dateien die Modul-Syntax zu der von ECMAScript und transpilieren Sie das Projekt erneut
- Ersetzen Sie alle require-Aufrufe durch entsprechende import-Anweisungen. Dabei benötigen Sie die folgenden drei Import-Varianten:

```
import * as express from 'express';  
import { MongoClient } from 'mongodb';  
import tasks from './routes/tasks';
```

- Ersetzen Sie die letzte Zeile (`module.exports = router;`) in `tasks.ts` durch

```
export default router;
```

5. Ändern Sie den Konfigurationsparameter `target` zu ES2017 und transpilieren Sie die Anwendung erneut. Welche Unterschiede entdecken Sie bei den generierten JavaScript-Dateien?

```
express-tm  
db  
node_modules  
src  
  public  
  routes  
    tasks.ts  
  views  
    app.ts  
package.json  
tsconfig.json
```

Übung 2

Aufgaben: Typannotationen



1. Ändern Sie den Konfigurationsparameter `strict` zu `true` und beheben Sie die Compile-Fehler durch Hinzufügen von Typannotationen
 - Fügen Sie hierzu den Dateien `app.ts` und `tasks.ts` folgenden Import hinzu, damit Sie den Typ `Db` verwenden können
2. Zeigen Sie anhand von drei Stellen im Quellcode, dass der Typ inferiert werden kann
3. Demonstrieren Sie die Vorteile statischer Typinformationen in Bezug auf Tippfehler und Code Completion am Quellcode der `express-tm`-Anwendung

```
import { Db } from 'mongodb';
```

Übung 3

Aufgaben: Interfaces und Enum-Typen



1. Erstellen Sie ein Interface Task in der Datei `src/models/task.ts` mit folgenden Eigenschaften

Eigenschaft	Erläuterung
<code>id</code>	ID der Aufgabe (UUID)
<code>title</code>	Aufgabentitel (Zeichenkette)
<code>createdAt</code>	Zeitstempel der Erzeugung der Aufgabe (Zahl)
<code>status</code>	Bearbeitungsstatus der Aufgabe (Enum-Typ mit den String-Konstanten 'open' und 'done')

- Für die Eigenschaft `status` ist ein Enum-Typ `TaskStatus` zu definieren (ebenfalls in der Datei `task.ts`)
2. Verwenden Sie das Interface und den Enum-Typ in der Datei `tasks.ts`

Übung 4

Aufgaben: Klassen

1. Erstellen Sie in der Datei `src/models/task.dao.ts` eine Klasse `TaskDAO` mit folgenden Methoden

Methode	Erläuterung
<code>create(task: Task, cb: any) { }</code>	Speichert das Objekt in der DB und ruft anschließend den Callback <code>cb</code> auf: <code>cb(err, task);</code>
<code>findAll(cb: any) { }</code>	Liest alle Aufgaben, sortiert nach dem Erstellungszeitpunkt, aus der DB aus und übergibt diese mit dem Callback-Aufruf: <code>cb(err, tasks);</code>
<code>update(task: Task, cb: any) { }</code>	Aktualisiert die Aufgabe in der DB und ruft anschließend den Callback auf: <code>cb(err, task);</code>
<code>delete(id: string, cb: any) { }</code>	Löscht die Aufgabe mit der angegebenen ID in der DB und ruft dann den Callback auf: <code>cb(err, deleted);</code>

Die Implementierung erfolgt analog zu den DB-Aufrufen in der Datei `src/routes/tasks.ts`.

`deleted: true`, wenn das Dokument gelöscht wurde
(`result.deletedCount > 0`)

- Der Konstruktor soll das übergebene DB-Objekt in der Dateneigenschaft `db` ablegen

Übung 4

Aufgaben: Klassen



2. Fügen Sie folgende Zeile der Datei `app.ts` an geeigneter Stelle hinzu

```
app.locals.taskDAO = new TaskDAO(db);
```

3. Aktualisieren Sie schrittweise die Datei `tasks.ts`, sodass dort nicht mehr direkt auf die DB zugegriffen wird, sondern die zugehörige Methode des `TaskDAO`-Objekts aufgerufen wird

- Überprüfen Sie zwischendurch die Funktionalität der Anwendung

Übung 5

Aufgaben: Generics



1. Erstellen Sie in der Datei `src/models/generic.dao.ts` eine Klasse `GenericDAO` als Verallgemeinerung von `TaskDAO` wie folgt
 - Erstellen Sie in der gleichen Datei ein Interface `Entity`, das Eigenschaftssignaturen für die Eigenschaften `id` und `createdAt` enthält
 - Die Klasse soll einen Typparameter `T` für den Typ der Dokumente besitzen
 - Schränken Sie dabei die Menge der erlaubten Typen für den Typparameter `T` so ein, dass `T` zuweisungskompatibel zum Typ `Entity` ist
 - Der Konstruktor soll neben dem DB-Objekt den Namen der Collection übergeben bekommen

Übung 5

Aufgaben: Generics



2. Ersetzen Sie in den Datei `app.ts` und `tasks.ts` die Verwendung der Klasse `TaskDAO` durch die Klasse `GenericDAO`
3. Erstellen Sie in der Datei `src/models/generic.dao.ts` ein Interface `DAOCallback<T>`, dass die Callback-Funktionen generisch typisiert
 - Der Typparameter `T` steht dabei für den Typ des 2. Parameters der Callback-Funktion

Übung 6

Aufgaben: Fortgeschrittene Typen



1. Ersetzen Sie im Interface Task den Enum-Typ durch einen geeigneten Union-Typen mit String-Literaltypen für die Eigenschaft status
 - Passen Sie anschließend die Verwendungsstellen (Datei tasks.ts) entsprechend an
 - Vergleichen Sie beide Ansätze (Enum-Typ vs. Union-Typ) miteinander. Wo sehen Sie Vorteile beim zweiten Ansatz?
2. Setzen Sie in der Datei tasks.ts in der Funktion für die Behandlung von PATCH-Anfragen ein Partial des Typs Task ein
 - Welche Änderung ist zudem erforderlich, damit der Aufruf der update-Methode des DAOs keine Transpilierfehler erzeugt?