

## 1 Übung 2 - JavaServer Pages

Die zweite Übung soll es Ihnen ermöglichen, JavaServer Pages ein wenig kennenzulernen. Versuchen Sie einmal, die im Folgenden beschriebene Webanwendung umzusetzen. Auf Moodle finden Sie die Datei zoo.zip. Sie enthält bereits ein Maven-Projekt mit der grundlegenden Infrastruktur, so dass Sie sich ganz auf die Webanwendung konzentrieren können.

Das Projekt enthält die folgenden Dateien:

- Die Modellklassen Animal und Compound,
- die Klasse Database, welche die Datenbank mithilfe von statischen Methoden simuliert,
- ein Verzeichnis image mit den Bildern
- die Klasse Zoo-Controller, welche den Spring-Controller für die Koordination der Webanwendung enthält und in der Sie Ihre Navigationsmethoden und in der Sie Ihre Navigationsmethoden einbauen müssen,
- pom.xml mit allen benötigten Abhängigkeiten.

Sie dürfen natürlich auch gerne das gesamte Projekt selbst erstellen und müssen meine Klassen nicht verwenden. Schauen Sie sich aber auf jeden Fall den Code und die JavaDoc-Kommentare kurz an.

Abgabe: 19.07.2020

Laden Sie bitte das Maven-Projekt als zip-Datei im Moodle hoch.

Die folgenden Screenshots sollen nur eine ungefähre Vorgabe sein. Sie dürfen natürlich gerne auch mit JSP herumexperimentieren und noch weitere Funktionen einbauen. Ob Sie Scriptlets oder JSTL verwenden, ist Ihnen überlassen. Generell gilt die Empfehlung, dass man so weit wie möglich auf Scriptlets verzichten sollte.

Kleiner Tipp: Behalten Sie immer im Kopf, dass Sie eine Webanwendung mit dem zustandslosen HTTP entwickeln. Das heißt, dass Sie alle Daten, die Sie benötigen, immer in den Request stecken müssen. Denken Sie daran, dass Ihnen dazu bspw. versteckte Felder zur Verfügung stehen. Sämtliche Aktualisierungen können durch Aktualisierung der gesamten Seite geschehen. Sie benötigen also keine JavaScripte u.ä.

Das Programm soll einen Spaziergang durch einen Zoo simulieren. Als erstes kann der Benutzer ein Gehege auswählen, das er gerne besuchen möchte (Abbildung ??).

Hat er kein Gehege ausgewählt, erhält er eine Fehlermeldung. Im gesamten Programm sollen natürlich keine NullPointerExceptions auftreten (Abbildung 1).

# Willkommen im Zoo

## Welches Gehege möchten Sie besuchen?

### Schauen Sie auf unsere Karte

☐ Schlangen ☐ Afrika ☐ Vögel ☐ Pinguine

Gehe zum Gehege

Sie haben kein Gehege ausgewählt!

Abbildung 1: Auswahl eines Geheges

Hat der Benutzer ein Gehege ausgewählt, kommt er als nächstes zu den Tieren. Es gibt Tiere, die sich versteckt haben, und solche, die zu sehen sind. Der Anwender kann von den Tieren, die sich nicht versteckt haben, eines auswählen und zu ihm gehen. Auch hier gilt natürlich, dass eine Fehlermeldung angezeigt werden soll, wenn der Benutzer kein Tier wählt (Abbildung 2).

Hat der Benutzer ein Tier ausgewählt, werden ein Bild und der Hunger des Tiers angezeigt. Die Bilder sind in dem vorgegebenen Rumpf-Projekt bereits vorhanden. Der Name des Bilds ist in der Eigenschaft `image` der Modellklasse `Animal` abgelegt. Wenn ein Tier Hunger hat, kann der Benutzer es füttern. Der Hunger ist ein einfaches `int`-Attribut, das einen Wert zwischen 0 und 10 annehmen kann. Die Methode `feed()` der Klasse `Animal` verringert den Hunger um eins (Abbildung 3).

Wenn das Tier keinen Hunger hat, verschwindet die Schaltfläche und eine Meldung erscheint stattdessen. Für die gesamte Navigation innerhalb der Anwendung sollen Schaltflächen zur Verfügung stehen, damit der Benutzer nicht die Navigationsschaltflächen des Browsers benutzen muss (Abbildung 4).

# Sie sind hier: Vögel

**Sie sehen diese Tiere:**

☐ Adler [Gehe zum Tier](#)

**Diese Tiere verstecken sich:**

- Kondor
- Papagei
- Strauß

[Zurück zur Karte](#)

Abbildung 2: Fehlermeldung, wenn kein Gehege ausgewählt wurde

**Sie beobachten Zebra und Zebra beobachtet zurück.**



**Hunger: 9**

[Füttern](#)

[Zurück zum Gehege](#)

Abbildung 3: Auswahl eines Tiers zum Besuchen

**Sie beobachten Giraffe und Giraffe beobachtet zurück.**



Das Tier Giraffe hat keinen Hunger.

[Zurück zum Gehege](#)

Abbildung 4: Beobachten eines Tiers

**Sie beobachten Zebra und Zebra beobachtet zurück.**



Das Tier Zebra hat keinen Hunger.

[Zurück zum Gehege](#)

Abbildung 5: Tier wurde gefüttert und hat keinen Hunger mehr