

**Komponentenbasierte Entwicklung SoSe 2015**

**Übungsblatt 2: JEE/JSF/Komponentensysteme**

Bearbeitung in der Übung KW 17

Falls Ihnen weitere Informationen fehlen, recherchieren Sie im Web.
---

**Aufgabe 2-1 (JSF vs. JEE)**

**Vorbemerkung:**

- Wir bearbeiten die Aufgabe im *Timeboxing*-Verfahren. Bearbeiten Sie die Aufgabe immer nur soweit, wie vorgegeben. Melden Sie sich, wenn Sie mit einem Schritt fertig sind.
- Die einzelnen Schritte sollen im Paar bearbeitet werden. Die Paararbeit sollte sich wie folgt gestalten:
  - Bei Teilaufgaben, in denen Sie etwas lesen sollen, trennen Sie sich. Jeder liest für sich allein.
  - Bei Teilaufgaben, in denen Sie Fragen beantworten sollen, tun Sie sich bitte zusammen. Finden Sie gemeinsam eine Antwort. Machen Sie sich Notizen zu Ihrer Antwort.

**Ziel:**

- Kennenlernen des Oracle Tutorials zu Java EE7 (*The Java EE Tutorial*). Hierbei handelt es sich um eine „offizielle“ Anleitung zu Java EE in der Version 7 (aktuelle Version).
- Vertiefen des bisher Gelernten anhand des Tutorials.

**Die Aufgaben:**

- a) **Was ist der Unterschied zwischen der *Java Platform, Standard Edition (Java SE)* und der *Java Platform, Enterprise Edition (Java EE)*?** Um diese Frage zu beantworten lesen Sie bitte den einführenden Abschnitt unter [http://en.wikipedia.org/wiki/Java\\_Platform,\\_Enterprise\\_Edition](http://en.wikipedia.org/wiki/Java_Platform,_Enterprise_Edition). Folgen Sie nötigenfalls Links auf dieser Seite.
- Beantworten Sie folgende Fragen:
- I. Kann man Java EE unabhängig von Java SE verwenden?
  - II. Was sind die drei Haupteinsatzgebiete Java EE?
  - III. Welchen beiden zentralen Prinzipien folgt Java EE?
  - IV. Eines dieser beiden Prinzipien haben wir noch nicht erörtert. Welches ist das und was bedeutet es?
  - V. Was ist eine *Multitier architecture*?
  - VI. In welcher Form bildet eine auf JSF basierende Applikation eine *Multitier architecture*?
  - VII. Was ist das sogenannte *Web profile*?
- b) **Was sind Java EE Container?:** Öffnen Sie das *Java EE Tutorial* zu Java EE unter <https://docs.oracle.com/javaee/7/tutorial/index.html>. In diesem Tutorial wird der volle Umfang von Java EE Technologien behandelt. Hierzu gehören neben JSF viele weitere

Technologien, wie die Java Persistence API (JPA), Java Message Service (JMS), Java API for XML Web Services (JAX-WS), Enterprise JavaBeans (EJB) – nicht zu verwechseln mit den JavaBeans, die wir schon kennengelernt haben –, usw. Das ist viel mehr, als wir auch nur ansatzweise in unserer Veranstaltung behandeln können.

Öffnen Sie den Teil *Part I Introduction*, dann *Chapter 1, "Overview"* und zuletzt **Java EE Containers** (sie sollten jetzt bei <https://docs.oracle.com/javaee/7/tutorial/overview004.htm#BNABO> sein). Lesen Sie den Text soweit wie nötig, um folgende Fragen zu beantworten:

- I. Wofür sind in Java EE Container da?
  - II. Welche unterschiedlichen serverseitigen Container-Typen gibt es? Welchen der beiden Typen kennen wir schon? Wofür ist er laut dieser Dokumentation verantwortlich?
  - III. Was ist der Unterschied zwischen einem *Java EE Server* und einem Container?
  - IV. Haben wir mit dem Tomcat 7 einen Java EE Server oder einen Container im Einsatz?
- c) Bleiben Sie im *Java EE Tutorial* und gehen Sie zum Abschnitt *Part III The Web Tier* und dort zu Chapter 6, "Getting Started with Web Applications" (<https://docs.oracle.com/javaee/7/tutorial/partwebtier.htm#BNADP>): Was ist der Unterschied zwischen *Presentation-oriented Web Application* und *Service-oriented web Application*?
- d) Wechseln Sie zu *Web Application Lifecycle* (<https://docs.oracle.com/javaee/7/tutorial/webapp002.htm#BNADU>): Welche Schritte müssen beim Entwickeln einer Web-Applikation (mit Java) durchlaufen werden? Überlegen Sie bei jedem einzelnen Schritt: Was ist im Falle von JSF zu tun?
- e) **Nun noch etwas zu JSF:** Wechseln Sie zu *16 Configuring JavaServer Faces Applications* und dort zu Using Annotations to Configure Managed Beans (<https://docs.oracle.com/javaee/7/tutorial/jsf-configure001.htm#GIRCH>): Lesen Sie Absatz 16.1.1 Using Managed Bean Scopes: Was bedeutet `FlowScoped` grob? Was bedeutet `Dependent` grob?
- f) Lesen Sie nun [http://www.tutorialspoint.com/jsf/jsf\\_managed\\_beans.htm](http://www.tutorialspoint.com/jsf/jsf_managed_beans.htm): Fällt Ihnen ein entscheidender Unterschied zu der Erklärung unter Punkt e) auf? Wie lässt sich dieser Unterschied erklären?

## Aufgabe 2-2 (Komponentensysteme)

### Vorbemerkung:

- Tun Sie sich mit einem anderen Paar zusammen um diese Aufgaben zu bearbeiten.
- Lesen Sie zuerst die komplette Aufgabe.
- Recherchieren Sie nicht im Web nach der Lösung der Aufgabe! Sie nehmen sich (und Ihren Gruppenpartnern) die Möglichkeit, etwas zu lernen.
- Klären Sie zu Anfang, welche Informationen Sie einholen müssen, damit Sie mit der gemeinsamen Bearbeitung beginnen können – z.B. zur Funktionsweise von Unix-Pipes.

- Nehmen Sie sich einen festen Zeitraum vor, in dem jedes Mitglied allein nach den notwendigen Informationen recherchiert.
- Tragen Sie danach Ihre Rechercheergebnisse zusammen und bearbeiten die Fragestellung.
- Machen Sie sich in der Gruppe Notizen, so dass ein Mitglied Ihrer Gruppe das Ergebnis vortragen kann.

**Ziel:** Eigenschaften von Komponentensystemen verstehen.

Die Aufgabe besteht im Beantworten einer einzigen Frage: Kann man die Verkettung von Unix-Pipes als die Realisierung eines Komponentenmodells sehen?

Klären Sie die Frage anhand der in der Übung vorgestellten Kriterien an Komponentensystemen.