

---

# SEP/MP 2015

## – Aufgabenblatt 3 –

*spätester Abgabetermin: 29. Mai 2015 16 Uhr*

---

*Dieses Blatt richtet sich Teilnehmer des **Software-Entwicklungsprojekts**, wie auch an Teilnehmer des **Modellierungspraktikums**.*

### Ziel des Blattes

Auf dem letzten Aufgabenblatt wurde die Analysephase mit der Erstellung eines Analysemodells abgeschlossen. Auf diesem Aufgabenblatt wird nun die Entwurfsphase begonnen. Dabei wird das Analysemodell erweitert, konkretisiert und um technische Aspekte ergänzt.

Ihr Entwurfsmodell soll die Verbindung zu den verwendeten Frameworks bzw. zu den verwendeten APIs aufzeigen. Außerdem wird für Konzepte, die bisher nur fachlich definiert waren, sich so aber technisch nicht 1:1 umsetzen lassen, eine technische Lösung gesucht. Oftmals kommen dabei Entwurfsmuster (*Design Patterns*) zum Einsatz. Dies ist besonders daher interessant, da hier Server und Client getrennt werden.

### Aufgaben

1. Erstellen Sie um technische Aspekte verfeinerte Klassendiagramme, die die statische Struktur der Software veranschaulichen.
2. In Ihrer Software soll die Netzwerkkommunikation mittels RMI<sup>1</sup> umgesetzt werden. Achten Sie darauf, dass dies bereits in Ihren Modellen ersichtlich ist.
3. Zeigen Sie mit Hilfe einiger Sequenzdiagramme, wie die folgenden Szenarien technisch realisiert werden sollen:
  - a) Anmelden eines Teilnehmers
  - b) Erstellen eines neuen Raums
  - c) Beginnen einer neuen Spielpartie (Karten austeilen, etc.)
  - d) Eine komplette Spielpartie (sinnvoll zusammenfassen)

### Hinweise

1. Wenn Ihr Modellierungstool Beschränkungen hinsichtlich der Gestaltung der Diagramme aufweist, sodass Sie von dem Abweichen müssen, was in der SE2 gelehrt

---

<sup>1</sup>[http://openbook.rheinwerk-verlag.de/java7/1507\\_12\\_001.html](http://openbook.rheinwerk-verlag.de/java7/1507_12_001.html)

wurde, schreiben Sie das bitte dazu.

2. Wenn Sie absichtlich von dem abweichen, was in der SE2 gelehrt wurde, ist das auch kein Problem, solange Sie das sinnvoll begründen.
3. Das Analysemodell passt vielleicht gerade noch in ein Diagramm. Für das Entwurfsmodell sollten Sie mehrere Diagramme verwenden. Eine Möglichkeit ist, pro Paket ein separates Diagramm zu erstellen.
4. Mittels Stereotypen lassen sich bestimmte Informationen leicht an Klassen „anheften“. So ist es beispielsweise möglich, Remote-Interfaces durch einen Stereotyp `«remote»` zu kennzeichnen, anstatt jedes Mal einen Generalisierungspfeil zu `java.rmi.Remote` zu zeichnen. Auf diese Weise können Sie Ihre Diagramme übersichtlicher gestalten.
5. Bedenken Sie, dass bei RMI alle Daten, die zwischen Client und Server ausgetauscht werden, entweder `Remote` oder `Serializable` sein müssen.
6. Sehen Sie sich die *Java Collection API*<sup>2</sup> an. Sie werden einige dieser Klassen bzw. Interfaces brauchen.
7. Wenn Sie von dem abweichen, was Sie in der Analyse spezifiziert haben, dokumentieren Sie das. Passen Sie das Analysemodells entsprechend an. Bedenken Sie, dass gewisse Änderungen, wie beispielsweise das Aufteilen von Funktionalität von einer Klasse in mehrere, normale Verfeinerungen darstellen können und demnach nicht zwangsweise eine Abweichung vom Analysemodell sind.

---

<sup>2</sup><http://docs.oracle.com/javase/7/docs/api/java/util/Collections.html>