

# Übersicht und Rahmenbedingungen zur Wahlpflichtveranstaltung „Programmierung von Client/Server-Anwendungen“

*Prof. Dr.-Ing. habil. Andreas Schmietendorf*

Professur Wirtschaftsinformatik/Systementwicklung, HWR Berlin – FB II  
Privatdozentur Software-Engineering, OvG Universität Magdeburg – FIN

## **1. Übersicht zur Vorlesung**

Die Programmierung von Client/Server-Anwendungen ist innerhalb des industriellen Umfelds seit vielen Jahren fest etabliert. Für angehende Wirtschaftsinformatiker gilt es die grundlegenden Konzepte, welche sich hinter dem Client/Server-Paradigma verbergen, zu kennen und die implizierten Vor- und Nachteile möglicher Architekturentscheidungen einzuschätzen. Die Lehrveranstaltung vermittelt einen praxisorientierten Überblick zu ausgewählten Inhalten der Programmierung von Client/Server-Anwendungen. Zur Demonstration vielfältiger Implementierungsaspekte wird auf "low level" APIs der Programmiersprache Java, den Funktionsumfang des Java Enterprise Edition Framework (kurz JEE) bzw. alternativer Technologien (JS/HTML5/CSS) eingegangen. Begleitet wird die Vorlesung durch projektbezogene Übungen die den vermittelten Stoff im Rahmen einer teamorientierten Bearbeitung vertiefen sollen. Abgeschlossen wird das Fach durch eine Projektarbeit, die dafür geltenden Rahmenbedingungen finden sich in den folgenden Kapiteln des Flyers.

### Inhalte der Vorlesung:

#### Grundlagen von Client/Server-Systemen

- Ziele verteilter Systeme und Verteilungstransparenz
- Client/Server-Architekturen und Kommunikation
- Einsatz von Services

#### Verwendung von Applications Programmiers Interface

- Einsatz servicespezifischer APIs
- Anbindung von Datenbanksystemen mit JDBC, JPA und Co
- Generische Middleware-Mechanismen (Java-RMI und CORBA)

#### Funktionsumfang der JEE-Architektur

- Web Container (Servlets, JSP, Applets, POJOs und JSF)
- Komponentenmodelle (JavaBeans, Enterprise JavaBeans, CDI-Beans)
- Entwicklungs- und Laufzeitumgebungen (u.a. Apache Tomcat, JBoss)

## Moderne Webanwendungen im Front End

- Grundlagen REST-basierte Architekturen
- Idee der Single Page Applications (kurz SPA)
- Basistechnologien (JS, HTML5, Frameworks)

## Pflichtübungen zur Vorlesung

- Übung 1 (Verwendung der Brainstorming- und Metaplantentechnik),
- Übung 2 (Netzwerkprogrammierung mit Java),
- Übung 3 (Java-Datenbankanbindung mit Hilfe von JDBC und Co),
- Übung 4 (JSP/JSF-Komponenten oder JS/HTML5-Framework),

## Optionale Übungen - Exkurse:

- Übung A (Generic Middleware - Java RMI),
- Übung B (Java Persistence API - Java JPA),
- Übung C (JDBC-basiertes Arbeiten mit Apache Derby).

Die optionalen Übungen dienen der Unterstützung im Zusammenhang mit der Projektbearbeitung, insbesondere der Aufgaben 4 und 5 innerhalb der Übung 3. Die Aufgaben 1 bis 3 der Übung 3 werden dadurch nicht hinfällig!

## **2. Übersicht zur Projektarbeit**

Die zu erbringende Prüfungsleistung im Modul WI403 „Programmierung von Client/Server-Anwendungen“ besteht in der semesterbegleitenden Durchführung eines Softwareentwicklungsprojekts in Gruppen von maximal 4 Teilnehmern (Ausnahmen bedürfen der Abstimmung mit dem Dozenten). Jede Gruppe erhält eine Aufgabenstellung, die entweder auf der Grundlage eines eigenen Vorschlages oder aber durch ein vom Dozenten vorgeschlagenes Thema festgelegt werden kann. Grundsätzlich muss der fachliche Hintergrund der umzusetzenden Lösung eine Anforderung aus dem Bereich der Wirtschaftsinformatik reflektieren. Typische Beispiele für ein solches System wären ausgewählte Funktionen der folgenden Typen von Softwareanwendungen:

- Verwaltung von Kundenaufträgen
- System zur Erstellung von Rechnungen
- System zur Verwaltung des Personalbestands

Die Projektbearbeitung ist während des Semesters (teilweise im Kontext der durchzuführenden Übungen 3 und 4) durchzuführen. Grundsätzlich hat jede Gruppe die folgenden Teilleistungen zu erbringen:

- 1. Projektreview A:** Verwaltung persistenter Daten (z.B. mit Hilfe des JDBC-Interfaces und entsprechender Java-Klassen oder dem JPA). Der

entsprechende Vortrag (Analyse, Konzept/Entwurf & Durchstich) geht mit 25% in die Bewertung ein.

2. **Projektreview B:** Umsetzung der Präsentationsschicht-GUI (z.B. mit Hilfe der JSP/JSF-Technologie in Java oder als SPA mit Hilfe von JS/HTML5). Der entsprechende Vortrag (Analyse, Konzept/Entwurf & Durchstich) geht mit 25% in die Bewertung ein.
3. **Projektbericht:** Jede Gruppe liefert einige Wochen nach dem Ende des Semesters einen Projektbericht ab. Der Projektbericht wird mit 50% der Gesamtleistung bewertet und enthält Dokumentationen zu allen Phasen (Analyse, Konzept/Entwurf, Durchstich, Test) der Softwareentwicklung.

Bei dem zu entwickelnden System muss es sich um eine mehrstufige Client-/Server-Anwendung auf der Grundlage der JEE- bzw. JS/HTML-Technologie handeln. In jedem Fall kann die zu entwickelnde Anwendung mit Hilfe eines Browsers genutzt werden, daher gilt es eine Schicht für den „Presentation Layer“ mit Hilfe eines Web Servers (z.B. JSP - Apache) respektive Browsers (bei SPA-Ansätzen) auszuprägen. Darüber hinaus enthält die Anwendung einen „Application Logic Layer“ (z.B. Servlets – Apache oder EJBs - JBoss) und einen „Resource Management Layer“ welcher typischer zur Anbindung eines Datenbanksystems genutzt wird (z.B. RDBMS wie MySQL oder auch eine NoSQL-Datenbank wie z.B. die CouchDB oder auch die MongoDB).

### **3. Inhaltliche Bewertungskriterien**

Im Rahmen der Aufgabenbearbeitung muss deutlich zwischen den Phasen Analyse, Entwurf, Codierung und Test unterschieden werden können. Diese verschiedenen Phasen können aber durchaus auch (teilweise) parallel bzw. unter Berücksichtigung agiler Prinzipien und Techniken bearbeitet werden.

Sämtliche Ergebnisse der Projektbearbeitung sind festzuhalten und sofern sinnvoll modellorientiert (Schwerpunkt UML-Notation der OMG) zu dokumentieren. Als Dokumente sind mindestens zu erstellen:

- User Stories (ggf. Pflichtenheft), das die fachlichen Anforderungen an die zu erstellende Client/Server-Anwendung beschreibt,
- ein Analysemodell, das das fachliche Problem darstellt (umgangssprachlich und evtl. auch mit geeigneten UML-Diagrammen untersetzt),
- ein Entwurfsmodell, das die technische Umsetzung mit Hilfe des C/S-Paradigmas beschreibt (z.B. durch Klassen- und Ablaufdiagramme bzw. Deployment-Diagramme),
- der dokumentierte Quellcode, sowie verwendete Konfigurationsdateien (z.B. XML-Deployment-Deskriptoren),
- Testprotokolle, die die durchgeführten Entwickler-, Integrations- und Abnahmetests dokumentieren,

- sowie ein Nutzerhandbuch und eine Installationsanweisung für alle implementierten Client- und Server-Komponenten.

Diese Dokumente sollten in einem Band zusammengefasst werden, müssen aber dennoch deutlich gegeneinander abgrenzbar sein.

#### **4. Organisatorische Bewertungskriterien**

Es ist ein Projektplan zu erstellen, der deutlich zeigt, welche Phasen des Projekts wann geplant sind und welche Aufgaben von den Gruppenmitgliedern übernommen werden. Außerdem ist auf einer gesonderten Übersicht (ein DIN A4-Blatt) zu notieren, welches Gruppenmitglied welche Aufgaben übernommen hat. Beide Dokumente sind den Projektdokumenten hinzuzufügen. Auf dem Titelblatt der Projektdokumente müssen deutlich die Matrikelnummern der Gruppenmitglieder hervorgehen.

Die Abgabe der Übungsprotokolle (Übungen 1-4) und projektbezogenen Dokumente (inkl. Quellcodes und Testdaten) erfolgt mit Hilfe Moodle-System. Der Quellcode kann auch mit Hilfe eines projektspezifischen GitHub-Repositories zur Verfügung gestellt werden. Als Abgabetermin wird der xx. Mai 2022 festgesetzt. Im Rahmen der Abgabe sind sämtliche Dokumente und Quellcodes, sowie eventuell genutzte Testdaten zu berücksichtigen.

#### **5. Formale Bewertungskriterien**

Die gestellte Aufgabe ist in einer Projektgruppe von maximal 4 Teilnehmern zu bearbeiten. Es ist bei der Projektplanung darauf zu achten, dass jeder Teilnehmer für eine ausreichend große Arbeitsmenge eingeplant wird. Die folgende Aufzählung zeigt neben einer Bewertungsaufteilung auch eine ungefähre Verteilung des Arbeitsumfangs für ein Softwareprojekt:

- Form (10%):
  - o Rechtschreibung, Layout, Übersicht, allgemeiner Eindruck
- Analyse (10%):
  - o Vollständigkeit, Bearbeitungstiefe, Notation
- Entwurf (20%):
  - o Genauigkeit, Übersichtlichkeit, Geschick, Notation
- Codierung (30%):
  - o Funktionsumfang, Komplexität, Verständlichkeit, Wartbarkeit, Wiederverwendbarkeit
- Test (10%):
  - o Abdeckung, Vollständigkeit, Dokumentation
- Benutzer-/Installationshandbuch (20%):

- Verständlichkeit, Übersicht

Bei einer Gruppenstärke von 4 Mitgliedern sollten pro Mitglied mindestens 20% Gesamtanteil pro Mitglied übernommen werden. Sofern dieser Anteil nicht eingehalten wird, kommt es zu entsprechenden Abschlägen bei der Bewertung. Liegt der Anteil unter 15% (bei 4 Mitgliedern) wird die Leistung des betroffenen Projektmitarbeiters als nicht bestanden gewertet.

Die Bewertung der Projektarbeit erfolgt bei erkennbar gleichmäßiger Arbeitsverteilung und vergleichbarem Engagement der einzelnen Mitglieder möglichst einheitlich. Bei erkennbaren Abweichungen in der Leistung erfolgt jedoch auch eine differenzierte Bewertung!

## **6. Zeitplan der virtualisierten Vorlesung**

offen