

Prüfungsformalitäten

Umfang des Portfolios

Die Portfolioprüfung beinhaltet die beiden Vorlesungen „Webprogrammierung“ und „Verteilte Systeme“ aus dem dritten und vierten Semester. Das Prüfungsportfolio setzt sich deshalb aus folgenden Bestandteilen zusammen, die wie folgt verrechnet werden:

3. Semester: Webprogrammierung

Artefakt	Gewichtung
Einleitung	1
Lernkontrollen in Moodle	1
Programmierprojekt	4

4. Semester: Verteilte Systeme

Artefakt	Gewichtung
Lernkontrollen in Moodle	1
Untenstehende Aufgaben	4
Reflexion	1

Regelung zur Gruppenarbeit

Mit Ausnahme des Programmierprojekts im dritten Semester sind alle Artefakte in Eigenleistung zu erstellen. Die Artefakte müssen den persönlichen Wissensstand widerspiegeln und soweit wie möglich einen individuellen Lösungsweg erkennen lassen, um gewertet werden zu können.

Export der Lernkontrollen in Moodle

Nach Abschluss des Semesters erhalten alle Studierenden ein PDF-Dokument mit den exportierten Lernkontrollen aus Moodle. Dieses sollte ausgedruckt und in die Portfoliomappe eingeklebt werden. Wir bitten deshalb darum, mit der Abgabe der Portfoliomappe zu warten, bis das PDF-Dokument verschickt wurde.

Abgabe der Portfoliomappe

Für „Verteilte Systeme“ muss eine eigene Portfoliomappe abgegeben werden. Quellcodes müssen nicht ausgedruckt werden. Allerdings müssen die Unterlagen deshalb für die Bewertung zusätzlich in Moodle hochgeladen werden. Abgabefrist ist Montag, 18. Mai 2020.

Aufgabenstellung

Aufgabe 1: Grundlagen verteilter Systeme

a) In den Folien des Grundlagenkapitels sind mehrere Strategien für den Umgang mit Fehlern in verteilten Anwendungen genannt. Die Fehlersituationen können dabei rein technischer oder auch fachlicher Natur sein. Suchen Sie sich zwei der genannten Strategien aus und erklären Sie anhand einem selbst gewählten Fallbeispiel wie eine davon auf einen technischen und die andere auf einen fachlichen Fehler angewendet werden kann. Jede Beschreibung sollte mindestens zehn Sätze umfassen und folgende Punkte erläutern:

- Beschreibung der Fehlersituation
- Umstände, unter denen der Fehler auftreten kann
- Maßnahmen zum Umgang mit dem Fehler

b) Betrachten Sie eine verteilte Anwendung aus Ihrem beruflichen oder privaten Umfeld und erläutern Sie, wie diese zwei der in den Folien genannten Transparenzarten erfüllt. Erklären Sie weiterhin, wie diese oder eine andere Anwendung zwei weitere Transparenzarten nicht erfüllt. Jede Transparenzart sollte mit mindestens fünf Sätzen behandelt werden.

c) Geben Sie in mindestens zehn Sätzen das CAP-Theorem in eigenen Worten wieder. Zeigen Sie dabei anhand eines selbst gewählten, nicht in den Folien beschriebenen Beispiels, wie Einschränkungen bei einer der drei Bedingungen die Einhaltung der anderen beiden sicherstellen können.

d) Recherchieren Sie im Internet die beiden Begriffe „logische Uhr“ und „Lamport-Uhr“ und beschreiben Sie mit jeweils fünf oder mehr Sätzen, wie diese funktionieren.

Aufgabe 2: Entwicklung von REST-Webservices

a) Erklären Sie in eigenen Worten die wesentlichen Merkmale von REST-Webservices. Gehen Sie insbesondere darauf ein, wie REST-Webservices die Besonderheiten von HTTP ausnutzen und welche Funktion die gängigen HTTP-Verben in Zusammenhang mit bestimmten URLs haben. Erklären Sie auch, nach welchen Prinzipien die URLs hierfür strukturiert sein sollten.

b) Entwerfen Sie einen HATEOAS-fähigen Webservice mit zwei oder mehr miteinander verknüpften Collections. Die Collections sollten hierfür in einer 1:n-Beziehung, die sich auch in den URLs niederschlägt, stehen. Dokumentieren Sie folgende Aspekte:

- Das Datenmodell des Webservices in geeigneter, grafischer Form
- Die Bedeutung und den Aufbau aller möglichen URLs des Webservices
- Für jede URL, die zulässigen HTTP-Verben und was diese bewirken
- Für jede Collection eine beispielhafte Antwortstruktur beim Abruf einer Ressource

Achten Sie insbesondere beim letzten Punkt darauf, das HATEOAS-Prinzip zu berücksichtigen. Das gewählte Fallbeispiel darf nicht denen aus den Folien oder der Vorlesung entsprechen.

c) Implementieren Sie eine einfache Version des eben entworfenen Webservices in Java mit dem Spring-Framework. Falls Ihr Datenmodell mehr als zwei Entitäten besitzt, genügt es, wenn die Anwendung zwei miteinander verknüpfte Entitäten umfasst. Reichen Sie folgende Artefakte als Lösung ein:

- Den kommentierten Quellcode aller Java-Klassen und Interfaces, die zur Implementierung der Spring-Anwendung notwendig sind.

- Einen Screenshot der vollständig aufgeklappten Ordner- oder Projektstruktur. Erklären Sie mit wenigen Worten zu jeder Datei ihre Aufgabe und Bedeutung.
- Einen oder mehrere Screenshots, aus denen die Funktionsweise des Webservices hervorgeht. Die Screenshots können mit Postman oder einer vergleichbaren Anwendung erstellt werden und sollten mindestens die HTTP-Anfragen und Antworten zum Anlegen eines Datensatzes und danach zum Abruf einer Gesamtliste zeigen.

d) Zeigen Sie, wie der eben entwickelte Webservice entweder in Java oder in JavaScript aufgerufen werden kann. Geben Sie hierfür den Quellcode an, um zunächst einen Eintrag der übergeordneten Entität abzurufen und anhand der empfangenen Antwortdaten eine abhängige Entität abzurufen.

Aufgabe 3: Cloubasierte Microservices

a) Wie unterscheiden sich moderne, für den Cloubetrieb entwickelte Anwendungen von herkömmlichen Webanwendungen? Recherchieren Sie hierzu den Begriff „12-Faktor-App“ und erklären Sie insbesondere die Faktoren „Prozesse“, „Nebenläufigkeit“ und „Einweggebrauch“. Wie tragen diese Faktoren dazu bei, die Skalierbarkeit einer Anwendung zu erhöhen?

b) Lesen Sie auf der Webseite von Martin Fowler die Beschreibung des [CircuitBreaker-Patterns](#) und beantworten Sie anschließend folgende Fragen darüber:

- In welchen Situationen werden CircuitBreaker benötigt und welche Probleme lösen sie?
- Welche Zustände nehmen CircuitBreaker in der Regel ein und wie verhalten sie sich jeweils?
- Warum sollte der Zustand der CircuitBreaker im Produktivbetrieb überwacht werden?

c) Warum besitzen die meisten Microservice-Architekturen einen Reverse Proxy als zentralen Gateway-Server? Welche Aufgaben erfüllt dieser typischerweise und wie trägt er dazu bei, die nachträgliche Erweiterbarkeit zu erhöhen?

d) Microservices lassen sich häufig in technische und fachliche Microservices einteilen. Zeigen Sie anhand dem in der Vorlesung besprochenen Beispiel „Notebook as a Service“, welche technischen Microservices eine Anwendung neben dem Gateway-Server häufig besitzt. Fertigen Sie hierfür jeweils einen Screenshot der dazugehörigen Weboberfläche an und erklären Sie seine Aufgabe daran.

Aufgabe 4: Reflexion

Schreiben Sie einen mindestens einseitigen Reflexionsbericht für das Portfolio, in dem Sie zu folgenden Fragen Stellung beziehen. Verwenden Sie hierfür die in Moodle bereitgestellte Dokumentenvorlage mit den vorgegebenen Formatierungseinstellungen und achten Sie darauf, dass ein neuer Absatz nur durch einmal ENTER-Drücken erzeugt wird.

1) Inwiefern sind Ihnen verteilte Anwendungen schon früher im privaten und beruflichen Alltag begegnet und in wie weit verstehen Sie nun besser, wie diese funktionieren? Nennen Sie mindestens ein Beispiel und nehmen Sie dabei Bezug auf die Definitionen im Grundlagenkapitel.

2) In welchem Zusammenhang werden Sie im Beruf voraussichtlich an der Konzeptionierung, Erstellung, Einführung, Dokumentation, Beratung, Schulung oder Vertrieb von verteilten Anwendungen oder Webanwendungen beteiligt sein? Um was für Anwendungen handelt es sich dabei und welche Rolle werden Sie vermutlich übernehmen?

Wahlweise können Sie hier neben dem vierten zusätzlich auch auf das dritte Semester eingehen.