

Braindump Vesys MSP FS17

Closed Book:

Aufgabe 1: Multiple Choice

Richtig/falsch mit Fragen über 2. Unterrichtsteil, vorwiegend, wenn nicht sogar ausschliesslich Fragen über RMI.

Aufgabe 2: RMI Aufgabe

Codebeispiel von einer Client-Server Verbindung. Server implementiert eine Servicefactory, gibt 2 Service Objekte zurück (Service ist ein normales Objekt, welches eine Zahl beinhaltet und eine Methode um diese Zahl hochzuzählen). Eines der Objekte wird in der Factory exportiert, das andere nicht). Auf dem Client wird dann das exportierte Objekt geholt, inkrementiert, noch einmal geholt und inkrementiert und dann ausgegeben. Das selbe wird dann noch einmal mit dem nicht exportierten Objekt gemacht.

Aufgabe a) war nun die Ausgabe hinzuschreiben.

Aufgabe b) Falls Programm einfach 30 Minuten laufen gelassen wird, ist dann Server und/oder Client noch am Laufen? Mit Begründung.

Aufgabe c) musste man die Proxy und normalen Objekte in ein Schema einzeichnen, wie es aussieht nach der Ausgabe.

Aufgabe 3: JMS

Client und Server Anwendung. Echo. Server liest ständig Nachrichten aus Queue und gibt auf Konsole aus was er geholt hat, und und gibt nochmal aus was er geschickt hat („S << „ + msg und dann „S >> „+msg).

Client versendet „Hello“ und gibt aus was er geschickt hat und liest dann mit receive eine Nachricht aus worauf er beendet.

Man musste nun drei Szenarien angeben, wie eine Ausgabe aussehen kann.

Aufgabe 4: Callbacks

Frage zu 5 Technologien, ob Callbacks möglich sind oder nicht. Man musste Stichwortartig begründen:

- http
- RMI
- JMS
- Websockets
- Akka

Open Book

Aufgabe 5: Websockets

Distributed HashTable

Netz aus Nodes, jeder Node hat eine eigene HashTable. Nodes können sich bei einem beliebigen Node anmelden und dieser teilt dann seinen bekannten Nodes mit, dass diese sich auch bei diesem Neuen Node anmelden sollen. Bei getAufruf wird einfach aus der lokalen HashTable geholt, bei put Aufruf wird der neue Wert an alle Nodes geschickt und auch in deren HashTables gespeichert.

Man musste nun:

- Konzeptionell die Nachrichten aufzeigen, die bei einem Anmeldevorgang eines neuen Nodes verschickt werden mussten
- Einige Methoden und Felder in der Klasse erzeugen damit die Implementation funktionieren kann.
- Die OnMessage, OnClose und onOpen Methode implementieren.
- Man musste noch beantworten ob es zu Inkonsistenz kommen kann mit dieser Implementation.

Aufgabe 6 Akka

Philosophenproblem: 5 Aktoren, bei welchen jeder einfach seinen nächsten kennt, Beim Start eines Philosophen wird ein „Timer“ angeschmissen, der den Aktor nach einer Weile hungrig werden lässt. Es gibt noch einen zweiten „Timer“, der dem Philosophen nach einer Zeit sagt, dass er aufhören soll zu Essen. Ein Aktor kann nur essen, wenn er eine Gabel hat, diese wird durch ein Objekt „Token“ dargestellt, welches weitergereicht wird falls der Aktor, der sie gerade hat entweder aufhört zu essen oder gar noch nicht hungrig ist.