

7. Übung „Nebenläufige und verteilte Programmierung“

Abgabe in der Vorlesung oder bis zum 17. Dezember in Raum 704

Aufgabe 17

6 Punkte

Eine Kommunikation bei RMI besteht immer aus einer Anfrage und einer Antwort. Auch wenn die Methode keinen Rückgabewert (void) liefert. Schreiben Sie ein Program mit dem Sie zeigen, daß der Client auch für Methoden ohne Rückgabewert auf die Antwort des Servers wartet. Durch die zusätzlichen Verwendung von Threads ist es auch möglich Nachrichten abzuschicken, ohne auf ein Ergebnis zu warten. Wie können Sie diese Kommunikationsvariante durch die Definition spezieller Klassen/Interfaces unterstützen? Überlegen Sie insbesondere auch, was Sie bei Kommunikationsfehlern (Remote Exceptions) machen können.

Aufgabe 18

13 Punkte

Reimplementieren Sie den Chat unter Verwendung von RMI.

- a) Überlegen Sie genau, welche RMI-Server-Objekte registriert werden müssen und welche RMI-Objekt-Referenzen in Nachrichten weitergegeben werden können.
- b) Die programmierten Chats sind zwar robust gegen Ausfälle von Klienten. Sie sind aber nicht robust bezüglich des Ausfalls des Servers. Überlegen Sie sich eine neue Struktur, bei der kein ausgezeichneter Server mehr existiert, sondern alle Klienten gleichberechtigt miteinander kommunizieren. Der Chat soll solange existieren, bis sich kein Klient mehr im Chatraum befindet. Überlegen Sie sich auch einen Registrierungsmechanismus, der das Nadelöhr eines Servers verhindert, aber dennoch die Existenz mehrerer unabhängiger Chats erlaubt.

Alternativ können Sie diese Aufgabe auch mit Socketverbindungen lösen.

- c) An welchen Stellen in Ihren Programmen ist das Warten auf die Beendigung des Methodenaufrufs überflüssig? Verwenden Sie die in Aufgabe 17 definierten Klassen/Interfaces um die entsprechenden Aufrufe zu ersetzen.