



Nebenläufiges Programmieren

Sommersemester 2008

Serie 14 Bonusserie

9. Juli 2008

Ausgabetermin: 9. Juli 2008

Abgabe: 18. Juli 2008 (11:00)

Diese Serie soll in Einzelarbeit gelöst werden! Auch Abschreiben lassen führt zur Disqualifikation.

Aufgabe 1 (10 Punkte) Schreiben Sie ein Java-Programm, daß das Stable-Marriage Problem löst. Dieses Programm soll das Problem in einem verteilten System lösen. Benutzen Sie dazu Java mit der RMI-Technologie. Es sind drei jar-Files abzugeben:

- **MatchMaker.jar** - Ein Aufruf der Main Klasse des Jar-Files soll den Matchmaker starten, bei der Registry anmelden und dann auf die Kunden warten. Die Main Klasse soll sich in dem Packet **de.uniKiel.informatik.softtech.concurrency.server** befinden (analog zum Interface).
- **Male.jar** - Ein Aufruf der Main Klasse des Jar-Files soll einen männlichen Kunden erzeugen, der sich bei der Registry Zugang zum Matchmaker verschafft und sich bei diesem anmeldet. Die Main Klasse soll sich in dem Packet **de.uniKiel.informatik.softtech.concurrency.client** befinden (analog zum Interface).
- **Female.jar** - Wie Male.jar nur für einen weiblichen Kunden. Die Main Klasse soll sich in dem Packet **de.uniKiel.informatik.softtech.concurrency.client** befinden (analog zum Interface).

Nutzen Sie für die Kommunikation zwischen dem Matchmaker und den Kunden bzw. zwischen den Kunden untereinander ausschließlich die Interfaces, die auf der Homepage zur Vorlesung zur Verfügung gestellt werden.

Nutzen Sie ein Properties-File **parameters.properties** im gleichen Verzeichnis wie das jeweilige jar-File, um folgende Parameter an das Programm zu übergeben:

java.rmi.server.codebase Der HTTP-Server auf dem die notwendigen Klassen zur Verfügung gestellt werden.

registry.server Der Server auf dem die Registry läuft.

registry.port Der Port über den die Registry erreichbar ist.

java.security.policy Ein Security-Policy für die Programme.

service.name Der Name unter dem der Matchmaker bei der Registry angemeldet wird.

N Die Anzahl der erwarteten männlichen bzw. weiblichen Kunden.

Nutzen Sie für das Properties-File die XML-Funktionen.

Folgender Programmablauf soll berücksichtigt werden.

- Der Matchmaker wird zuerst gestartet und bei der Registry registriert.
- Nur der Matchmaker wird bei der Registry angemeldet.
- Jeder Kunde ist ein eigenständiger Prozeß.
- Jeder Kunde kann konzeptuell auf einem eigenen Rechner laufen.
- Jeder Kunde versucht sich zunächst bei der Registry zu registrieren. Er erhält dabei eine eindeutige ID.
- Sobald sich ein Kunde beim Matchmaker registriert, beginnt der Matchmaker die bestehenden Kunden und den neuen bekannt zu machen.
- Sobald ein Kunde von einem potentiellen Partner erfährt, versucht er ein Date mit diesem zu bekommen.
- Eine Bewertung des potentiellen Partners findet nach dem Date statt. Die Bewertungen der Dates sind für einen Kunden eindeutig. Der Kunde sendet die Bewertung an den Matchmaker.
- Zwei Kunden bewerten ein Date unabhängig voneinander - insbesondere können sie ein Date unterschiedlich bewerten.
- Delays sollen nur bei Dates verwendet werden, die Zeit soll maximal eine Sekunde betragen. ($t \leq 1s$).
- Sobald der Matchmaker die Bewertungen aller Kunden für alle Dates erhalten hat, löst er das Stable-Marriage Problem und teilt jedem Kunden seinen neuen Partner mit.

Das Programm soll aussagekräftig dokumentiert werden. Das Programm soll beim Lauf aussagekräftige Log-Nachrichten erzeugen. Entweder per Consolen-Ausgabe oder über einen Logger. Wer einen Logger nutzen möchte sollte sich mit Immo Grabe igb@informatik.uni-kiel.de abstimmen. Erklären Sie auch die Lösungsstrategie zum berechnen einer Stable-Marriage.

Das Programm soll auch auf den Suns in der Universität laufen. Testen Sie Ihre Abgabe auch dort, so daß sichergestellt ist, daß Sie auch in einer von Ihrer Entwicklungsumgebung unterschiedlichen Java-Umgebung läuft.

Nachfragen bitte an Immo Grabe igb@informatik.uni-kiel.de
Viel Erfolg.