Übungsblatt 6

Ausgabe: 29.11.2011
Abgabe: 09.12.2011

Aufgabe 1 (Kalender mit RMI): (20 Punkte)

Ziel dieser Aufgabe ist es, ein Kalender-Server-Client-System zu programmieren. Der Server soll von Clients Befehle entgegen nehmen, wie ein Event (Name, Anfangszeit, zugeordnete Person) hinzuzufügen, entfernen, ändern. Weiterhin soll es zwei Mechanismen geben, über die ein Client informiert wird, wenn ein Event startet.

a) (5 Punkte) Laden Sie sich das Framework für die Aufgabe von der Veranstaltungsseite herunter. Benutzen Sie zum Lösen der Aufgaben die vorgegebenen Interfaces. Das Manipulieren bestehender Interfaces ist nicht erlaubt.
Machen Sie sich mit dem RMI Tutorial vertraut.
Implementieren Sie das Server-Interface (bis auf *getNextEvent*, *RegisterCallback* und *UnregisterCallback*).
Hinweise:

- Ein Event soll für jeden Benutzer des Systems sichtbar sein. Dementsprechend darf es nur eine Event-Speicherstuktur geben.
- Der Server soll mehrere Anfragen gleichzeitig bearbeiten können. Denken Sie an geeignete Sperrsynchronisierung.

b) (5 Punkte) Implementieren Sie einen Client, der die Befehle (add, remove...) von der Konsole entgegen nimmt und diese dann auf dem Server ausführt.

c) (2 Punkte) Starten Sie in der Main des Servers ein *RMIregistry*, registrieren Sie ein Server-Objekt und (re-)binden Sie das Objekt.

Starten Sie mindestens einen Client, der sich durch Angabe von IP und Port zu dem Server verbindet.

Testen Sie die Funktionen für die Events. Es soll auch mit mehreren Clients gleichzeitig funktionieren.

d) (4 Punkte) Implementieren Sie die Methode getNextEvent auf dem Server. Diese soll den Aufruf des Clients blockieren, bis die aktuelle Uhrzeit der Startzeit eines

Events (für einen bestimmten Benutzer) entspricht. Dann soll sie das entsprechende Event zurückliefern.
Dafür ist es wichtig, dass Sie geeignete Datenstrukturen verwenden um die Events zu sortieren und eine ressourcen-schonende Implementierung für die Zeitmessung finden.

Benutzen Sie die Methode *getNextEvent* sinnvoll in ihrem Client und testen Sie - Achtung, auch hier muss das System mit mehreren Clients gleichzeitig funktionieren - evtl. hilft Ihnen passing the baton oder ein Monitor?

e) (2 Punkte) Jetzt soll der Aufruf ebenfalls anders herum möglich sein: Der Server gibt dem Client ein Signal, wenn ein Event startet. Sollten Sie nicht mit dem Observer-Pattern vertraut sein, informieren Sie sich.

In unserem Fall ist der Client der Observer des Servers (Observable). Implementieren Sie die Methoden RegisterCallback und UnregisterCallback und auf der Clientseite das Interface *EventCallback*.
Achten Sie beim Programmieren auf der Serverseite darauf, dass der Server nicht blockiert wird, während call() auf einem EventCallback aufgerufen wird.

f) (2 Punkte) In der Dokumentation soll beschrieben werden, welche Vor- und Nachteile die jeweilige Methode zur Benachrichtigung hat. Denken Sie dabei auch an Firewalls, Synchronisations-Aufwand, Erweiterbarkeit des Systems usw.
Welche Methode preferieren Sie?

optional: g) (2 Punkte) Benutzen Sie die Serialisierungs-Schnittstelle von Java um die Event-Daten des Servers persistent zu speichern.