

# Software-Technik I (WS 2003/04)

## 4. Übungsblatt

Gegeben ist eine auf Java's Remote Method Invocation basierende Client-Server-Anwendung, deren Architektur dem Broker-Muster folgt. Sie können die Anwendung wie folgt auf *einem* Rechner ausführen:

1. Übersetzen der Quelltexte:  
`javac *.java`
2. Erzeugen von Stub und Skeleton (`rmic` ist Bestandteil des Java SDK):  
`rmic -keep TimeServiceImpl`  
Es entstehen die Dateien `TimeServiceImpl_Stub.java` und `TimeServiceImpl_Skel.java` sowie die entsprechenden `.class`-Dateien.
3. Starten der RMI-Registry (`rmiregistry` ist Bestandteil des Java SDK):  
`start rmiregistry` (Windows) bzw. `rmiregistry &` (Unix)
4. Registrieren einer Instanz von `TimeServiceImpl` bei der RMI-Registry:  
`start java Register1` (Windows) bzw. `java Register1 &` (Unix)
5. Ausführen des `TimeServiceClient`:  
`java TimeServiceClient`
6. Abmelden der Instanz von `TimeServiceImpl` aus der RMI-Registry:  
`java Unregister`
7. Registrieren einer anderen Instanz von `TimeServiceImpl`:  
`start java Register2` (Windows) bzw. `java Register2 &` (Unix)
8. Ausführen des `TimeServiceClient`:  
`java TimeServiceClient`

Gehen Sie wie folgt vor, wenn Client und Server auf *unterschiedlichen* Rechnern laufen sollen:

- Ersetzen Sie in `TimeServiceClient.java`, `Register*.java` und `Unregister.java` den Servernamen `localhost` durch den Namen des Server-Rechners.
- Starten Sie die RMI-Registry auf dem Server-Rechner, und führen Sie das Registrieren und Entfernen der Instanzen von `TimeServiceImpl` auf dem Server-Rechner aus.

## 1. Aufgabe

Stellen Sie die Kommunikation der folgenden sieben Interaktionspartner bei der Ausführung der obigen Schritte 4 bis 8 durch ein Interaktionsdiagramm dar:

- Client (Instanz der Klasse `TimeServiceClient`),
- Stub (Instanz der Klasse `TimeServiceImpl_Stub`),
- Registry (Implementierung der Schnittstelle `java.rmi.Naming`),
- zwei Skeletons (Instanzen der Klasse `TimeServiceImpl_Skel`),
- zwei Server (Instanzen der Klasse `TimeServiceImpl`, die sich selbst registrieren und abmelden.)

Erstellen ein mit dem Sequenzdiagramm konsistentes Klassendiagramm mit den Klassen `TimeServiceClient`, `TimeServiceImpl_Stub`, `Registry`, `TimeServiceImpl_Skel`, `TimeServiceImpl`, `TimeService`. (Dabei fasst `Registry` mehrere Klassen des JDK zusammen, die gemeinsam den Namens-Service bereitstellen.)

**Letzter Abgabetermin:** 02.12.2003