



## Aufgabe – Chat-System mit Java RMI (10 Punkte)

Sie sollen ein Chat-System unter Verwendung von Java RMI implementieren. Das System besteht aus einem Server, an den beliebig viele Clients angeschlossen werden können. Das System soll nach bekanntem Prinzip arbeiten: Nachrichten werden auf einem Client eingegeben und über den Server an alle angeschlossenen Clients weiter geleitet.

Es gibt zwei prinzipielle Möglichkeiten, ein solches System zu realisieren:

- Die Nachrichten werden vom Server vorgehalten. Verbundene Clients überprüfen in zyklischen Abständen das Vorliegen neuer Nachrichten und rufen diese ab (Polling).
- Clients registrieren im Server jeweils ein sog. Callback-Objekt. Dieses beinhaltet Methoden zum Empfang und zur Anzeige von Nachrichten.

Verwenden Sie für Ihre Lösung den letztgenannten Ansatz und **protokollieren Sie Ihr Vorgehen mithilfe der Vorlage**. Es sind hierzu zwei Klassen von Remote-Objekten vorzusehen:

- Die Klasse **ChatServer** stellt nach außen eine Methode zur Verfügung, mit der sich Clients beim Server unter einer Referenz auf das Client-Callback-Objekt (**ClientProxy**) registrieren und de-registrieren können:

```
public interface ChatServer extends Remote {  
    public ChatProxy subscribeUser (ClientProxy handle)  
        throws RemoteException;  
    public boolean unsubscribeUser (ClientProxy handle)  
        throws RemoteException;  
}
```

Außerdem verteilt der Server die Nachrichten an die Clients. Überlegen Sie sich dazu eine passende Methode!

- ClientProxy** repräsentiert den Client im Server und unterstützt eine Methode zum Empfang von Nachrichten:

```
public interface ClientProxy extends Remote {  
    public void receiveMessage (String username,  
        String message) throws RemoteException;  
}
```

Die entsprechenden Implementierungsklassen realisieren die Schnittstellen-Methoden sowie weitere Methoden. Die Implementierung der Schnittstelle **ChatServer** erzeugt in der **main()** Routine ein neues Objekt von sich selbst und registriert das Objekt dann in der **RMI Registry** unter einem frei wählbaren Namen. Daneben benötigen Sie noch einen Client (GUI oder interaktive Shell). Dieser liest eine Referenz auf das **ChatServer**-Objekt aus der **Registry** aus und ermöglicht es einem User, sich unter Angabe eines Namens per **subscribeUser()** Methode beim Chat zu registrieren.

Testen Sie Ihr System, in dem Sie einen Chat mit mehreren Clients aufsetzen.



## Schaubild

