

Name:

Matr.Nr.:

---

Ziel-Repository: `svn://www.ssw.uni-linz.ac.at/2009S/PSW2/<kMatrNr>/branches/ex3/`

Abzugeben: Quellcode, auf Papier und elektronisch.

Abgabefrist: 18. 1. 2010

Tutor:

Punkte:

Maximal: 100 Punkte

---

## Verteilte Anwendungen: Networking und Remoting

### Aufgabe 1: MindMap Kollaborationsplattform

Implementieren Sie auf der Basis von einfacher Network-Programmierung (`java.net`) einen Server und entsprechende Clients, die es erlauben, dass mehrere verteilte Benutzer gemeinsam an einer MindMap arbeiten. Es sollen folgende Operationen unterstützt werden:

- Laden der aktuellen MindMap vom Server
- Sperren eines Knotens in der MindMap durch einen Benutzer (der Knoten und alle Unterknoten können nicht durch andere Benutzer geändert werden)
- Uploaden eines geänderten Knotens zum Server; der Knoten muss vom Benutzer gesperrt sein; Freigeben des gesperrten Knotens.

Entwerfen Sie entsprechende Protokolle, um die Operationen zu unterstützen und implementieren Sie diese unter Verwendung von Sockets und des Streaming-APIs. Die Protokolle sollen auch mit fehlerhaften Interaktionen umgehen können.

#### Hinweise:

- Wie das Marshalling und Streaming der Daten erfolgt, ist Ihnen freigestellt. Möglichkeiten sind über `Reader` und `Writer` und XML-Serialisierung oder `ObjectInputStream` und `ObjectOutputStream` und binäre Serialisierung von Objekten.
- Entwerfen und dokumentieren Sie zuerst die Protokolle, auf deren Basis die Kollaboration erfolgen soll.
- Vergessen Sie nicht, einen synchronisierten Zugriff auf die MindMap am Server zu implementieren
- Für den Datenaustausch benötigt man eine eindeutige Identifikation der Theme-Knoten in der MindMap. Erweitern Sie Ihre MindMap-Applikation, dass ein eindeutiger Zugriff möglich ist.

### Aufgabe 2: Remote MindMap

Implementieren Sie auf Basis von Java RMI eine verteilte MindMap-Applikation. Die interaktiven Clients sollen über RMI auf die MindMap auf einer Server-Applikation zugreifen. MindMap und Theme sind als Remote-Objekte auszuführen. Über Remote-Listener sollen die Clients von Änderungen der MindMap am Server verständigt werden.

Die RMI-Applikation soll die elementaren Operationen Hinzufügen, Löschen und Ändern von Themes unterstützen. Undo und Redo muss nicht unbedingt funktionieren!

#### Hinweise:

- Vergessen Sie nicht, einen synchronisierten Zugriff auf die MindMap-Daten am Server zu implementieren.
- Vergewissern Sie sich, dass alle Remote-Objekte `UnicastRemoteObject` erweitern.
- Achten Sie auf die Listener-Callbacks zur Client-Applikation.

### Punkte

- Aufgabe 1: 50 Punkte
- Aufgabe 2: 50 Punkte