**RMI Dokumentation Übung 1**

**Aufgabenstellung**

Erstelle mit Hilfe von RMI eine Java-Anwendung, welche es ermöglicht, im Client eine Zahl einzugeben, welche dann beim Server quadriert wird und anschließend am Client ausgegeben wird.

**Realisierung**

Ich habe mich entschieden, das Programm in zwei Projekte zu unterteilen, ein Projekt, in welchem der Server und der Namensdienst, die Registry, liegt und ein zweites, welches den Client simulieren soll. Da während der Realisierung des Projekts sehr viele Probleme aufgetreten sind, bei welchen die Lösung nicht sofort ersichtlich war, steckte ich viel Zeit in das Projekt, um die Aufgabenstellung zu realisieren.

**Ablauf des Programms**

Um die RMI-Anwendung zu starten, muss man darauf achten, dass Folgende Reihenfolge vom starten der drei Komponenten eingehalten wird:

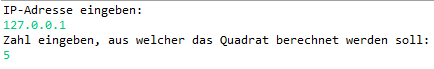
* Anfangs muss die Registry gestartet werden
* Als zweites muss der Server gestartet werden
* Als Drittes muss der Client gestartet werden

Bei meiner Anwendung wird die Registry automatisch gestartet, wenn der Server ausgeführt wird, da sie sich im Server befindet.

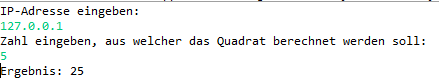
Wurden die Anwendungen richtig gestartet, verlangt die Clientanwendung auf der Konsole eine IP-Adresse, mit welcher sich der Client zum Server verbinden kann.

 Hier wird die IP-Adresse eingegeben, um sich mit den Server zu Verbinden. In diesem Fall 127.0.0.1, um sich mit den Localhost zu verbinden.

Danach befiehlt die Konsole eine Zahl einzugeben, welche quadriert wird.



nachdem man die Zahl bestätigt, wird das Ergebnis vom Server zurückgegeben.



Antworten:

b)Mit tasks kann ein Server verschiedene Aufgaben erledige, ohne geändert zu werden. Nachteile sind Problemme bei der Realisierung

1. Das Interface 'java.rmi.Remote' enthält Schnittstellen und Funktionen, welche erst RMI und Remote Objekte ermöglichen
2. Da ich Aufgabe 2.2 nicht erledigen konnte, kann ich für diese Frage keine Antwort geben
3. Server kann gleichzeitig Clients nacheinander abarbeiten