

**Protokoll**

**DEZSYS-04 "VERTEILTE OBJEKTE MIT RMI"**

**INSY**

**4CHIT 2015/16**

**Otahal Maro**

|  |  |
| --- | --- |
|  | **Version 1** |
| **Note:** | **Begonnen am 30. März 2016** |
| **Betreuer: Roschger** | **Beendet am 30. März 2016** |

Inhaltsverzeichnis

[1 Einführung 3](#_Toc447109994)

[1.1 Ziele 3](#_Toc447109995)

[1.2 Voraussetzungen 3](#_Toc447109996)

[1.3 Aufgabenstellung 3](#_Toc447109997)

[1.4 Hinweise 3](#_Toc447109998)

[2 Ergebnisse 4](#_Toc447109999)

[2.1 Erste Select Abfrage: 4](#_Toc447110000)

[2.2 Zweite Select Abfrage 5](#_Toc447110001)

[Zeitaufzeichnung 7](#_Toc447110002)

# Einführung

## Verteilte Objekte haben bestimmte Grunderfordernisse, die mittels implementierten Middlewares leicht verwendet werden können. Das Verständnis hinter diesen Mechanismen ist aber notwendig, um funktionale Anforderungen entsprechend sicher und stabil implementieren zu können.

## Ziele

Diese Übung gibt eine einfache Einführung in die Verwendung von verteilten Objekten mittels Java RMI. Es wird speziell Augenmerk auf die Referenzverwaltung sowie Serialisierung von Objekten gelegt. Es soll dabei eine einfache verteilte Applikation in Java implementiert werden.

## Voraussetzungen

* Grundlagen Java und Software-Tests
* Grundlagen zu verteilten Systemen und Netzwerkverbindungen
* Grundlegendes Verständnis von nebenläufigen Prozessen

## Aufgabenstellung

Folgen Sie dem offiziellen Java-RMI Tutorial, um eine einfache Implementierung des PI-Calculators zu realisieren. Beachten Sie dabei die notwendigen Schritte der Sicherheitseinstellungen (SecurityManager) sowie die Verwendung des RemoteInterfaces und der RemoteException.

Implementieren Sie ein Command-Pattern [2] mittels RMI und übertragen Sie die Aufgaben/Berechnungen an den Server. Sie können am Client entscheiden, welche Aufgaben der Server übernehmen soll. Die Erweiterung dieser Aufgabe wäre ein Callback-Interface auf der Client-Seite, die nach Beendigung der Aufgabe eine entsprechende Rückmeldung an den Client zurück senden soll. Somit hat der Client auch ein RemoteObject, welches aber nicht in der Registry eingetragen wird sondern beim Aufruf mittels Referenz an den Server übergeben wird.

## Quellen

* Verwende dazu die mysql-Version der schokoDB (ist im Kurs hochgeladen).
* Nutze Statements um sql injections zu vermeiden.
* Abzugeben sind ein Arbeitsprotokoll sowie das php Plugin

# Ergebnisse

problem:

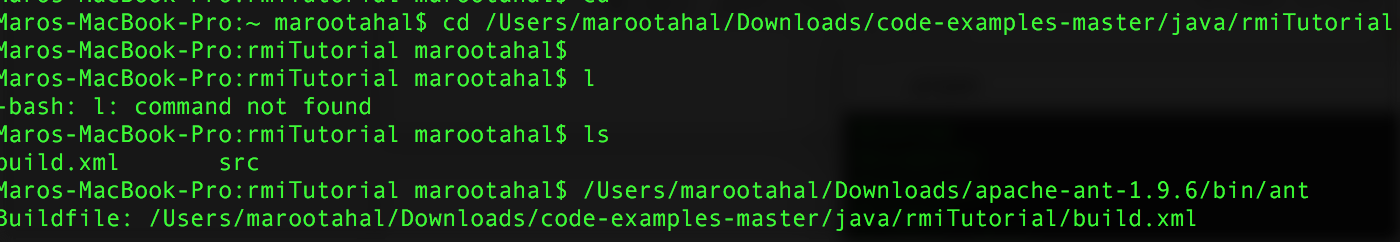
bei Mac ist der port 1099 von rmi Prozess schon selber genutzt daher muss man den port mit folgenden befehl löschen

lsof -i :1099

kill -9 PID\_number

befehl um den Prozess zu schließen

lsof -ti TCP:1099 | xargs kill



# Zeitaufzeichnung

|  |  |
| --- | --- |
| **Aufgabe** | **Dauer** |
| PhP Plugin schreiben | 1 h |
| Protokoll | 1 h |
| **Gesamt: 2 h** | |