
Laborprotokoll

DEZENTRALE SYSTEME

**Systemtechnik Labor
4CHIT 2015/16, Gruppe Z**

Steinwender Jan-Philipp

Note:

Betreuer: Broko

Version 1.0

Begonnen am 1. April 2016

Beendet am 1. April 2016

Inhaltsverzeichnis

1Einführung	3
1.1Ziele	3
1.2Voraussetzungen.....	3
1.3Aufgabenstellung	3
2Ergebnisse	4

1 Einführung

Initialisiertes Repository: <https://github.com/jsteinwender-tgm/SYT-Labor-Borko.git>

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.

1.1 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.

1.2 Voraussetzungen

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

1.3 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.

1.4 Quellen

[1] "The Java Tutorials - Trail RMI"; online:

<http://docs.oracle.com/javase/tutorial/rmi/>

[2] "Command Pattern"; Vince Huston; online:

<http://vincehuston.org/dp/command.html>

[3] "Beispiel Konstrukt für Command Pattern mit Java RMI"; Michael Borko;

online: <https://github.com/mborko/code-examples/tree/master/java/rmiCommandPattern>

2 Ergebnisse

Umgebungsvariablen ändern

Java.Policy

Pfad der java version C:\Program Files\Java\jre1.8.0_60\lib\security

Den pfad mit / slash nicht mit \ backslash. Den Pfad ins Projekt zeigen, nicht in src Verzeichnis

<http://docs.oracle.com/javase/8/docs/technotes/guides/security/PolicyFiles.html>

```
grant codeBase "file:/C:/Users/Jan-Philipp/TGM/eclipse/rmiTutorial/-" {  
    permission java.security.AllPermission;  
};
```

Der Client bereitet alles vor. Der Task ist binarisch. Übergibt es an den Server.

http://dita-ot.sourceforge.net/doc/ot-userguide13/xhtml/installing/windows_installingant.html

Ant installieren

Ant clean

Ant build

Ant engine

Am client:

Ant compute