

# Modul Verteilte Software WS 2022/2023

## Aufgabe 7

### RMI

1.

Ein netzfähiges Wörterbuch aus dem Übungsblatt 5 soll unter Verwendung von RMI (Remote Method Invocation) implementiert werden:

Das Wörterbuch soll zu einem deutschen Wort die Übersetzung z. B. in Klingonisch liefern (bzw. hinzufügen) und verfügt über vier Befehle, mit denen die auf dem Server gespeicherten Daten manipuliert werden können:

- **put** <de> <kli>: erstellt ein neues Werte Paar oder überschreibt ein existiertes Paar auf dem Server, z. B. **put genug mevyap**
- **get** <de>: fragt den entsprechenden Wert mit einem Schlüsselwort ab, z. B. **get genug**
- **delete** <de>: löscht ein existiertes Wertepaar mit dem angegebenen Schlüsselwort, z. B. **delete genug**
- **quit**: beendet die aktuelle Verbindung mit dem Server und speichert das Wörterbuch in eine Datei.

#### Programmierhinweise:

- Die entfernten Methoden sind in einem von Remote abgeleiteten Interface zu definieren. Alle Operationen des Remote-Interface (hier gibt es nur eine) können RemoteException werfen.
- Das Remote-Interface muss in einer von UnicastRemoteObject abgeleiteten Server-Klasse implementiert werden. Die Serverklasse benötigt einen Konstruktor, der auch ggf. RemoteException werfen kann. Um dem Remote-Objekt die entfernte JVM verfügbar zu machen, müssen in der main-Funktion des Servers folgende Schritte durchgeführt werden:
  - Registry erstellt werden:  
**LocaleRegistry.createRegistry(Registry.REGISTRY\_PORT);**
  - Das Remote-Objekt mit dem Registry-Namen verbunden werden:  
**Naming.rebind("Server",new Server());**
- Implementieren Sie die RMI Client Klasse, die nach der Verbindung mit dem Server (per **Naming.lookup("//localhost/Server")**) die Wörterbuchoperation auf dem Server ausführt und das Ergebnis auf Client-Konsole anzeigt.

2.

Erstellen Sie eine RMI-Anwendung, die serverseitig eine Methode zur Verfügung stellt, die mit Hilfe eines Polynoms Primzahlen produziert.

Client übergibt dabei an den Server ein Objekt, das die Polynom-Koeffizienten und die Anzahl der Primzahlen beinhaltet. Server liefert die Primzahlen als ein Array.

#### Hinweise:

Erstellen Sie für das Parameter-Objekt eine Klasse, die das Interface Serializable implementiert. Mit Euler-Polynom ( $n^2-n+41$ ) können die Primzahlen für  $n=0..39$  berechnet werden, mit Beyleveld-Polynom ( $n^4-97n^3+3294n^2-45458n+213589$ ) für  $n=0..49$