

# Voting

## 1. Allgemein:

Mittels RMI soll ein Votingsystem implementiert werden. Dem User soll mittels Konsoleneingabe möglich sein für einfache Ja/Nein – Fragen abzustimmen, sich eine Liste der Fragen generieren zu lassen und eine Statistik anzusehen.

Die Befehle lauten:

l	...listet alle Fragen auf
s	...zeigt die Abstimmungsstatistik an
v <Frage>	...stimmt für eine bestimmte Frage ab
q	...beendet das Programm

Zur Erinnerung:

Einlesen von Kommandozeile:

```
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
```

## 2. Server

Die Fragen sollen am Server gespeichert werden, wobei die Fragen selbst bei der Instanziierung des Servers aus einer Textdatei namens „Fragen.txt“ eingelesen werden sollen. Zu den Fragen soll außer der Frage selbst auch gespeichert werden können, wie viele Ja/Nein – Stimmen bisher abgegeben wurden. (Tipp: eigene Klasse verwenden (zb Question)).

Bietet Clients folgende Funktionen

- Liste der Fragen.
- Abstimmen für eine Frage.
- Datengrundlage für Statistikberechnungen liefern.

## 3. Client

Am Client selbst muss nur die Kommandozeilenfunktionalität implementiert werden.

- Auflistung der Fragen.
- Abstimmen für eine Frage.
- Auflistung des Wahlergebnis zu allen Fragen mit prozentueller Verteilung der abgegebenen Stimmen.

Geben Sie nach jeder Interaktion erneut die Liste mit den verfügbaren Befehlen aus. (siehe oben).

Lesen Sie dann immer die Zeile ein und überprüfen Sie, welcher Befehl ausgeführt werden soll und nutzen Sie dann die entsprechenden Methoden des Servers, um sich die jeweiligen Listen vom Server zu holen und am Bildschirm auszugeben.

Tipp: Wenn Sie zu einer Frage abstimmen wollen, lesen Sie zuerst die Zeile ein und fordern Sie danach den Benutzer zum Eingeben seiner Antwort auf.

Es bietet sich an einen StringBuffer zu verwenden um die ursprüngliche Frage zusammen zu setzen und dann mit dem eingegebenen String (Ja oder Nein) an die entsprechende Methode des Servers zu übergeben.