

# Verteilte Systeme — Übungsblatt 3

Prof. Dr. Oliver Haase

In dieser Aufgabe geht es darum, den Bully-Algorithmus zum Wählen eines neuen Anführers zu programmieren.

Zum einfacheren Ausführen und Testen sollen die einzelnen 'Prozesse' (im Sinne des Bully-Algorithmus) lokal als nebenläufige Threads ausgeführt werden. Dazu ist die Klasse **GlobalSystem** gegeben, die 10 *Bully-Prozesse* erzeugt und als Threads ausführen lässt, Verknüpfungen (Kommunikationskanäle) zwischen allen herstellt, und den Wahlprozess startet.

Ihre Aufgabe besteht darin, die fehlende Klasse **BullyProcess** zu programmieren. Dazu ist bereits die abstrakte Basisklasse **Process** vorgegeben, die Ihre Klasse **BullyProcess** erweitern soll. Wie Sie sehen, ist in **Process** die Methode **receiveMessage** definiert, die dazu verwendet wird, zwischen den Prozessen (Threads) Nachrichten auszutauschen. Konkret ruft ein Prozess  $P_1$ , der einem anderen Prozess  $P_2$  eine Nachricht schicken möchte, an  $P_2$  dessen **receiveMessage**-Methode auf. Dadurch, dass  $P_2$  die übergebene Nachricht in eine interne Warteschlange einreicht, wird entfernte, nachrichtenbasierte Kommunikation simuliert. In der Struktur **destinations** merkt sich jeder Prozess, welchen anderen Prozessen er Nachrichten übermitteln kann; diese Struktur simuliert also die Ausgangskanäle zu den empfangenden Prozessen.

Schließlich ist die Klasse **Message** gegeben, die sowohl für ELECT-Nachrichten als auch die Antworten darauf verwendet werden soll. Das Feld **uuid** ist dazu gedacht, einen Wahlvorgang eindeutig zu kennzeichnen, so dass ein Bully-Prozess, der mehrere ELECT-Nachrichten mit derselben **uuid** bekommt, entsprechend reagieren kann.

Schreiben Sie nun die fehlende Klasse **BullyProcess**. Gehen Sie dazu wie folgt vor:

- Implementieren sie die fehlende **run**-Methode. Begeben Sie in dieser Methode den Kontrollfluss in eine Schleife, innerhalb derer Nachrichten von der internen Warteschlange gelesen und verarbeitet werden. Dazu müssen Sie geeignet eingehende ELECT-Nachrichten quittieren, selbst ELECT-Nachrichten versenden und auf eingehende Antworten korrekt reagieren.
- Implementieren Sie außerdem die Methode **startElection**, mit deren Hilfe das **GlobalSystem** einen Auswahlvorgang von außen starten kann.

Viel Spass & gutes Gelingen!