

# Verteilte Systeme — Übungsblatt 2: Nebenläufigkeit

Prof. Dr. Oliver Haase, Dr. Thomas Zink

**Hinweis:** Bitte lesen Sie vor Bearbeitung dieser Übung unbedingt das *Laborskript zur Nebenläufigkeit in Java*.

## Aufgabe 1: Nebenläufiger Aqualife-Broker

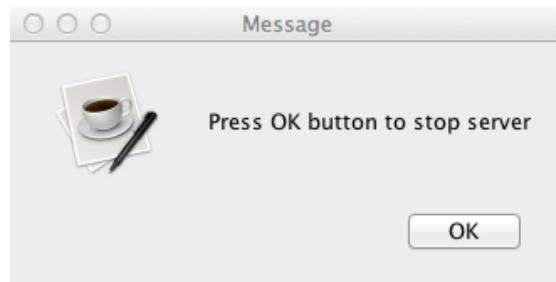
Der Broker vom Übungsblatt 1 behandelt alle Anfragen seriell in nur einem Thread. In dieser Aufgabe sollen Sie den Broker so erweitern, dass er zur Bearbeitung der eingehenden Anfragen einen Thread-Pool konstanter Größe verwendet. Benutzen Sie dazu das Java Executor-Framework und gehen Sie konkret wie folgt vor:

- Erstellen Sie eine (innere) Klasse `BrokerTask`, die die Verarbeitung und Beantwortung von Nachrichten übernimmt.
- In der `broker`-Methode des Brokers erzeugen Sie nun für jede eingehende Nachricht eine neue Instanz von `BrokerTask` und übergeben Sie diese dem `ExecutorService` zur Ausführung.
- Beachten Sie, dass die Klasse `BrokerTask` Zugriff auf die Client-Liste des Brokers benötigt, und dass diese konkurrierende Zugriffe synchronisiert werden müssen. Um parallele Ausführung von `Handoff-Requests` zu erlauben, verwenden Sie zur Synchronisation einen `ReadWriteLock` wie im Laborskript zur Nebenläufigkeit beschrieben.

## Aufgabe 2: `stopRequested`

In dieser Aufgabe sollen Sie den Broker so erweitern, dass er nicht in einer Endlosschleife auf eingehende Requests hört, sondern programmatisch beendet werden kann. Ändern Sie dazu die Schleife in der Methode `broker` so, dass der Broker Anfragen entgegen nimmt, bearbeitet und beantwortet, bis ein `stopRequested`-Flag gesetzt wird. Ergänzen Sie den Broker um einen weiteren Thread, der dem Benutzer über eine grafische Eingabemaske die Möglichkeit gibt, den Server zu beenden. Sobald der Benutzer die

entsprechende Eingabe gemacht hat, soll dieser Thread das `stopRequested`-Flag setzen. Für die Eingabemaske verwenden Sie am einfachsten die Methode `showMessageDialog` der Klasse `JOptionPane`. Sie sieht dann in etwa wie folgt aus:



### Aufgabe 3: Poison-Pill

Verwenden Sie nun die Poison-Pill-Technik, um den Broker beendbar zu machen. Laden Sie hierzu aus dem Moodle-Bereich der Arbeitsmaterialien die Jar-Datei `aqua-blatt2-poison-src.jar` herunter und importieren diese in Ihr Projekt. Die Datei enthält die Klassen `Poisoner` und `PoisonPill`. `PoisonPill` ist ein neuer Nachrichtentyp, der die Poison-Pill darstellt. Wird diese vom Broker empfangen, so soll er sich beenden. `Poisoner` ist ein spezieller Client, der die Poison-Pill an den Broker sendet, sobald der Benutzer auf 'OK' klickt. Testen Sie hiermit, ob Ihr Broker sich ordentlich beendet.

Viel Spass & gutes Gelingen!