

Verteilte Systeme — Übungsblatt 1

initiale Version der verteilten Aqualife-Anwendung

Prof. Dr. Oliver Haase, Dr. Thomas Zink

In dieser Übung sollen Sie die teilweise vorgegebene verteilte Anwendung Aqualife vervollständigen, die im *Laborskript zur Beispielapplikation* beschrieben ist. Bitte unbedingt zuerst dieses kurze Skript lesen!

Für diese Übung benötigen Sie mehrere Jar-Dateien, die Sie in Moodle unter *Arbeitsmaterialien* finden: Die Bibliothek `messaging.jar` enthält die Klassen `Endpoint` und `Message`, die eine nachrichtenbasierte Kommunikation zwischen Kommunikationsendpunkten anbieten. Fügen Sie diese Bibliothek Ihrem Projekt hinzu. Die Dokumentation der Klassen liegt als Javadoc in dem Archiv `messaging-javadoc.jar` vor.

Desweiteren benötigen Sie die Quellen des Projektes *Aqualife*. Laden Sie `aqua-blatt1-src.jar` herunter und importieren Sie das Archiv in Ihr Projekt.

Aufgabe 1: Broker

Das Archiv `aqua-blatt1-src.jar` enthält eine initiale Version des Aqualife-Klienten sowie das unvollständige Paket `aqua.blatt1.broker`, das nur die Klasse `ClientCollection` enthält. In dieser Aufgabe sollen Sie dieses Paket zu einem initialen, lauffähigen Aqua-Broker ergänzen. Dazu müssen Sie eine neue Klasse `Broker` wie folgt schreiben:

- Der `Broker` benötigt einen `Endpoint`, der auf Port 4711 hört.
- Der `Broker` muss eine Liste verfügbarer Clients führen. Verwenden Sie dazu die vorgegebene Klasse `ClientCollection`.
- In der `broker`-Methode soll in einer Endlosschleife blockierend auf Nachrichten gewartet werden. Ankommende Nachrichten müssen dekodiert und die im folgenden beschriebenen Methoden aufgerufen werden.
- Die `register`-Methode wird aufgerufen bei einem `RegisterRequest`. Der `Broker` vergibt eine neue ID, beispielsweise `"tank1"` für den ersten Client, `"tank2"` für den

zweiten, usw., trägt den neuen Client in die Client-Liste ein und antwortet ihm mit einer `RegisterResponse`-Nachricht.

- Die `deregister`-Methode wird aufgerufen bei einem `DeregisterRequest`. Der `Broker` entfernt den Client aus der Client-Liste.
- Die Methode `handoffFish` wird aufgerufen bei einem `HandoffRequest`. Der `Broker` ermittelt den betroffenen Nachbarn und gibt den `HandoffRequest` an diesen weiter.
- Die `main`-Methode instantiiert einen neuen `Broker` und startet die `broker`-Methode.

Testen Sie Ihren Broker, indem Sie mehrere `Aqualife` Klienten starten und die Fische beobachten. Diese sollten zwischen den verschiedenen Client-Fenstern hin und her schwimmen.

Aufgabe 2: Verteilen

Verteilen Sie Broker und Klienten auf unterschiedliche Maschinen, beispielsweise indem Sie gemeinsam mit einem oder mehreren Kommilitonen einen Broker und mehrere Klienten auf verschiedenen Rechnern laufen und miteinander kommunizieren lassen. Hierzu müssen Sie in der Klasse `aqua.blatt1.common.Properties` die Adresse und den Port des Brokers festlegen. Die Konstante `BROKER_NAME` wird noch nicht benötigt. Die Klasse `Properties` muss bei allen Klienten sowie dem Broker entsprechend angepasst werden.

Starten Sie die verteilte `Aqualife`-Anwendung. Die Fische sollten nun zwischen den Klienten der verschiedenen Maschinen hin und her schwimmen.

Viel Spass & gutes Gelingen!