

Übung 06: Client-Server

Abgabetermin: 11. 5. 2020, 8:15

Aufgabe	Punkte	Abzugeben über Moodle
Übung 6	24	Java-Programm in Zip-Datei, PDF-Datei für Korrektur

Übung 6: Client-Server für Kontaktverfolgung (24 Points)

In dieser Übung sollen Sie eine Client/Server-Applikation implementieren, um sich bewegende Entities zu verfolgen.

Implementieren Sie dazu folgende Kommunikation mit Client und Server:

- Der Client meldet sich beim Server an.
- Der Server antwortet, indem er dem Client eine eindeutige Id zuteilt.
- Der Client schickt laufend Positionsdaten zum Server
- Der Server prüft auf Kontakte und schickt erkannte neue Kontakte zum Client.
- Der Client empfängt die Kontakte und speichert diese.
- Der Client zeigt das Ende mit einer Terminate-Message an.
- Der Server bestätigt diese.

Hinweise:

- Am Server muss jeder Client in einem eigenen Thread behandelt werden.
- Am Client müssen das Schicken der Positionsdaten und das Empfangen der Kontaktdaten in zwei unterschiedlichen Threads erfolgen!

Besondere Anforderungen:

Teilung des Gebietes am Server:

Um die Kontaktsuche auch für größere Szenarien mit vielen Entities zu ermöglichen, soll am Server das Gebiet in Teilgebiete aufgeteilt werden. Die Entities müssen beim jeweiligen Teilgebiet angefügt werden. Bei Positionsänderungen müssen die Entities die Teilgebiete wechseln.

Im Gegensatz zur letzten Übung kann es hier günstiger sein, wenn Sie die Entities nur dann in ein Teilgebiet einfügen, wenn sie sich wirklich im Gebiet befinden. Sie müssen dann bei der Kontaktsuche auch die Entities der Nachbargebiete berücksichtigen.

Achten Sie besonders auf eine zuverlässige und effiziente Synchronisation der Zugriffe auf die gemeinsamen Daten. Überlegen Sie genau, wo wechselseitiger Ausschluss notwendig ist und welche Synchronisationsmechanismen geeignet sind.

Beenden der Client- und Server-Applikation:

Es soll möglich sein, dass die Client- und Server-Applikation durch eine Benutzereingabe beendet werden können, z.B. einfach durch Enter in der Konsole. Es müssen damit alle Threads beendet werden.

Test:

Testen Sie Ihr System mit einem Gebiet entsprechender Größe (`Constants.SIZE`) und einer Menge sich zufällig bewegendes Entities. Wählen Sie die Parameter so, dass es zu Kontakten kommt.

Download:

Im Download finden Sie eine Klasse `contact.WalkUtil` mit einer Methode `walk`, mit der die Position eines Entities zufällig ändert werden kann, und eine Methode `delay`, mit der man den aktuellen Thread eine zufällig bestimmte Zeit schlafen legen kann.

Hinweis:

Passen Sie eventuell die Klassen `Area`, `Entity` etc. aus dem Download zu Übung 5 entsprechend an.