

Aufgabe 6:

Aufgabenstellung:

Realisiere eine Client-Server Anwendung in Java. Der Server bietet zwei Funktionen (getRandom() und getTime()), die eine lange Berechnung simulieren sollen. Für diese Aufgabe sollten zwei Serverversionen erstellt werden. Eine sequenzielle Version, die nur einen Thread verwendet und somit die Clients nacheinander abarbeitet und eine parallele Version die mehrere Threads verwendet und die Clients parallel abarbeitet. Teste beide Server und dokumentiere deine Beobachtungen. Welche Einschränkungen und welches Leistungsvermögen haben die Server?

Implementierung:

Auf Ausgaben in der Konsole wurde verzichtet, da Ein- und Ausgabe mit JOptionPane erfolgt.

- Bei der parallelen Version lokal auf dem Rechner nicht Ideal, aber nur aufgrund von Testzwecken

Es wird eine Klasse Messages (erbt von Abstrakter Klasse) verwendet:

- Austausch zwischen Client und Server, speichert Befehl und Resultat der Berechnung

Bei der Verbindung wird Host und Port im Quellcode festgelegt (Bessere Testbarkeit, weil schneller ausführbar)

Version 1: Der Server verarbeitet die Clients sequenziell

Version 2: Der Server verarbeitet die Clients parallel

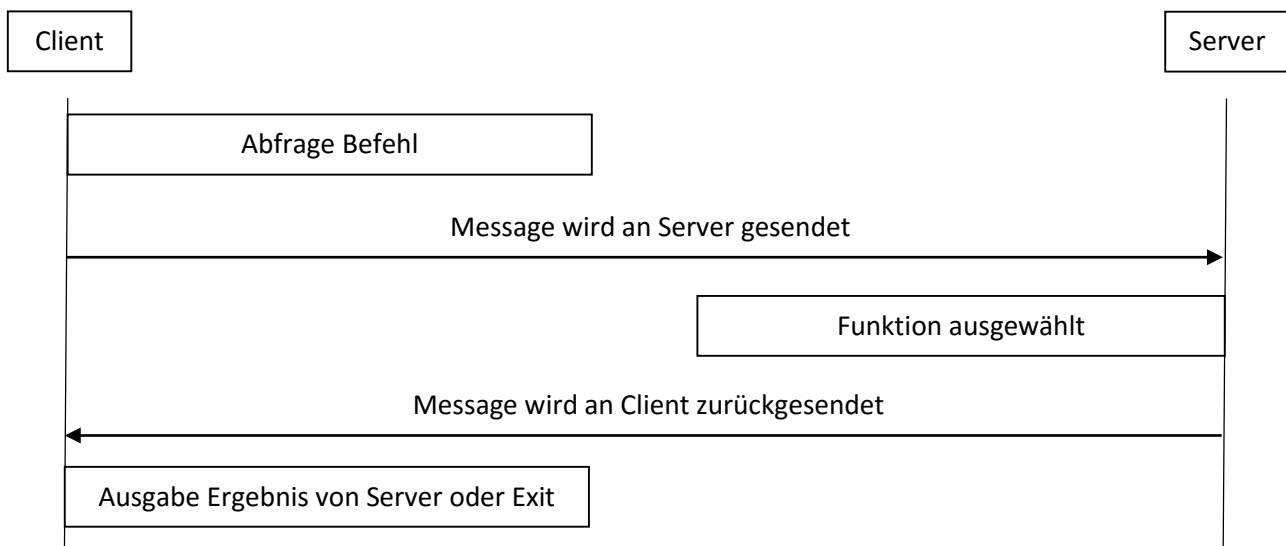
Mittels einer Schleife wird so lange weitergemacht bis der User exit eingibt

Server sind zwar meistens Leistungsstärker als die Clients, jedoch bei Überlastung des Servers ist die Berechnung/Kommunikation nicht mehr vorteilhaft.

Die wichtigsten Codestellen wurden im Quellcode kommentiert.

Ablaufdiagramm:

Bei beiden Versionen läuft der Vorgang so ab, nur dass Clients parallel oder sequenziell abgearbeitet werden:



Zustandsdiagramm:

