

AI-B.41: Verteilte Systeme

[Ü]

9

Bonusaufgabe // TCP + Threading

Ch. Wolf & T. Dumke | SG AI | FB 4 | HTW Berlin

Stand: WiSe 18/19

Urheber: Christopher Wolf & Tobias Dumke

Verwertungsrechte: keine außerhalb des Moduls

Inhalt

- Allgemeine Informationen
- Allgemeine Anforderungen
- Abgabeform/-frist
- Abgabehalt
- Aufgabe
 - » Client
 - » Server

Allgemeine Informationen

- praktischer Teil Voraussetzung zum Erzielen des Bonus
- zusätzlich notwendig zu 80% (8 von 10) der Übungen
- eine von drei praktischen Aufgaben auswählen
 - » UDP + Threading
 - » TCP + Threading
 - » RMI + REST API (Client)

Allgemeine Anforderungen

- Programmiersprache: Java
- Objektorientiert und Komponentenbasiert entwickeln
- sauberer Quellcode (*Quellcodequalität!*)
 - » **keine** Ein-Klassen- oder Alles-In-Main-Lösungen
 - » Quellcodeduplizierung vermeiden
 - » wenn möglich CCD Prinzipien
- IDE/Editor nach eigener Wahl
- Drittanbieter Bibliotheken sind gestattet
 - » begründen wieso Bibliothek X benutzt wurde

Abgabeform/-frist

- Abgabe erfolgt via E-Mail an entsprechende_n Dozent_in
- Abgabe bis 27.6.2019 in der Übung
- nicht fristgerecht eingereichte Abgaben werden nicht gewertet

Präsentation

- Vor- und Nachname
- Matrikelnummer
- Quellcode
- Präsentation der Anwendung
- Fragen beantworten

Aufgabe: Client

Entwickeln Sie eine Java-Anwendung (Client), bei welcher es möglich ist ein Datum (ohne Uhrzeit) anzugeben, um dann die Temperaturwerte der einzelnen Stunden (00:00 – 23:00 Uhr), sowie die Minimal-, die Maximal- und die Durchschnittstemperatur des Tages auf der Kommandozeile ausgeben zu lassen.

Die Wetterdaten werden über die TCP Socket API (Stream Sockets) eines von Ihnen entwickelten Servers abgefragt. Dabei handelt es sich um die 24 Temperaturwerte der einzelnen Stunden des angegebenen Tages.

Mögliche auftretende Fehler müssen von allen beteiligten Komponenten sinnvoll behandelt werden.

Aufgabe: Server [I]

Der *Server* soll ebenfalls in Java entwickelt werden und stellt die erwähnte **TCP Socket API** bereit. Er liest Temperaturwerte aus einer CSV-Datei ein, welche Sie zuvor angelegt haben. Es ist hier durchaus ausreichend, wenn die CSV-Datei für sieben Tage stündliche Temperaturwerte enthält.

Wenn der *Server* eine Anfrage für ein Datum erhält, zu welchem keine 24 Temperaturangaben existieren, wird ein Fehler an den Client zurückgegeben.

Zudem kann der *Server* Anfragen von verschiedenen Clients gleichzeitig bearbeiten. Dafür ist es notwendig den *Server* multi-threaded zu implementieren. Verwenden Sie hierfür das Java Interface *Runnable*.

Mögliche auftretende Fehler müssen von allen beteiligten Komponenten sinnvoll behandelt werden.

Aufgabe: Server [II]

Als alternative Datenquelle kann anstelle einer CSV-Datei eine freiverfügbare REST API verwendet werden, welche Wetterdaten zur Verfügung stellt.

Eine solche freiverfügbare REST API ist beispielsweise auf [OpenWeatherMap.org](https://openweathermap.org) zu finden.

AI-B.41: Verteilte Systeme

[Ü9]

Viel Erfolg!

Ch. Wolf & T. Dumke | SG AI | FB 4 | HTW Berlin

Stand: WiSe 18/19

Urheber: Christopher Wolf & Tobias Dumke

Verwertungsrechte: keine außerhalb des Moduls