

Fortgeschrittene Programmiertechniken

Prof. Dr. J. Pauli, Dipl.-Inform. J. Hoefinghoff, J. Kapitza, M. Cherubim

Übungsblatt 5

Seite 1 von 3

Auf dem letzten Aufgabenblatt haben Sie einen Einstieg in das Thema Threading erhalten. Auf diesem Übungsblatt wird das Thema Netzwerkkommunikation behandelt. Dabei ist es wichtig, dass bestimmte Aufgaben parallel ablaufen, daher sollten Sie ihr Wissen aus dem letzten Übungsblatt nutzen. Die Kodierung von String muss UTF8 nutzen.

Hinweise: Es reicht aus, wenn die Server alle Meldungen, bzw. Fehler auf der Konsole anzeigen.

Für beide gilt: die IP ist localhost oder auch 127.0.0.1.

Wenn der Service bereitgestellt wird, sollte zumindest eine Verbindung mit lokaler IP funktionieren.

Aufgabe 1

Update

Für die nächste Abgabe ändern Sie die Artifact ID in der pom.xml direkt unter der Gruppen ID von u3u4 auf u5u6. Und benennen (optional) Ihr Projekt mit der in Eclipse zu Verfügung gestellten Rename Funktion um. Testen sie mit **clean assembly:assembly** ob anschließend das zip mit dem Namen GRUPPE-u5u6-abgabe.zip erstellt wurde.

Aufgabe 2

UDP

Erweitern Sie die Klasse ViewCustomer ihres HardwareShops so, dass die Uhrzeit durch ein UDP-Paket von einem Server (Server-Port 6667) angefragt wird. Dazu sendet der Client (ViewCustomer) jede Sekunde ein UDP-Paket mit dem String 'time' an den Server (auf den Port 6667). Der Server liest das UDP-Paket und gibt auf der Konsole die IP und den Port des anfragenden Client aus. Nun erstellt der Server ein Antwortpaket und übermittelt in ihm die Uhrzeit des Servers. Die Uhrzeit wird von *java.util.Date* erzeugt und in einen String konvertiert. Der String sollte dann die Form 'TT/MM/JJJJ HH:MM:SS' besitzen (z.b. '31.12.2010 00:00:00'). Schauen Sie sich die Klasse *SimpleDateFormat* für diesen Zweck an. Während der Client jede Sekunde vom Server die Uhrzeit anfragt, darf die GUI nicht einfrieren! Ebenso sollen die Antworten des Servers nicht zum Einfrieren führen und mittels **SwingUtilities.invokeLater** in die GUI überführt werden.

Fortgeschrittene Programmiertechniken

Prof. Dr. J. Pauli, Dipl.-Inform. J. Hoefinghoff, J. Kapitza, M. Cherubim

Übungsblatt 5

Seite 2 von 3

Aufgabe 3

TCP Bestellungen

Erweitern Sie die Klasse `ViewCustomer` ihres `HardwareShops` so, dass es dem Nutzer möglich ist Bestellungen an ein Versandlager zu entsenden. Durch einen Klick auf den Kaufen-Button soll vom Benutzer in einem Dialog Name und Passwort abgefragt werden. Die Eingabe soll auf dem Server überprüft werden. Richten Sie die Authentifizierung so ein, dass lediglich der Nutzer `admin` mit dem Passwort `admin` eine Bestellung tätigen darf. Die Nutzerauthentifizierung kommt mit jedem Klick auf den Kaufen-Button und darf nicht zwischengespeichert werden.

- Um eine schnelle Lieferung zu ermöglichen, soll der `HardwareShop` seinem externen Versandlager die eingegangenen Bestellungen weiterleiten. Dazu sollen `Stream-Sockets` (Server-Port 6666) genutzt werden. Die Ausgabe kann entweder über die Konsole oder über eine grafische Oberfläche erfolgen. Ein mögliches Layout, wie der Versandhandel die Bestellungen anzeigen kann, zeigt [Abbildung 1](#).
- Denken Sie bei TCP-Verbindungen daran, `flush` zu verwenden. Die Verbindungen sollten nicht ständig neu aufgebaut und geschlossen werden.
- Achten Sie darauf, dass nur eine TCP Verbindung zum Server existiert. Es ist möglich mit mehreren Threads auf ein und demselben IO Stream zu arbeiten. Die Zugriffe müssen dabei kontrolliert ablaufen (Synchronisation).
- Schreiben Sie die Klasse `Warehouse`, welche auf ankommende Order-Objekte horcht und diese in einer geeigneten Datenstruktur speichert. In Ihrer Ausgabe soll die Klasse Informationen über jede neue Order ausgeben (Produkte, Stückzahl, Wert). Des Weiteren soll angezeigt werden, welche Produkte, in welcher Stückzahl insgesamt bestellt wurden und was für einen Gesamtwert die Bestellungen haben.
- Erweitern Sie ihren `HardwareShop` so, dass dieser alle eingehenden Orders an das `Warehouse` weiterleitet.

Info: Das `Warehouse` soll eine eigene `main`-Methode enthalten und keine Referenz auf das Model des `HardwareShops` besitzen.

INCOMING Thread, der sich um die eingehenden Nachrichten kümmert. Dieser liest die Daten ein und wertet die Informationen aus.

OUTGOING Thread, der sich um das Versenden von Nachrichten kümmert. Dieser bekommt eine Nachricht übergeben und sendet Sie an alle Beteiligten.

GUI Verwenden Sie für alle Interaktionen auf GUI-Elementen die Methode `SwingUtilities.invokeLater`, direkter Zugriff auf die GUI-Elemente kann dazu führen, dass die GUI kurzzeitig einfriert.

Der Server muss evtl. mehrere Threads erzeugen, um alle Clients parallel zu bearbeiten. Auf der Clientseite laufen demnach mindestens drei Threads und auf dem Server zwei Mal `X` Threads, wobei `X` die Anzahl der Clients ist. Eine übersicht über einen möglichen Aufbau finden Sie in [Abbildung 2](#).

Fortgeschrittene Programmiertechniken

Prof. Dr. J. Pauli, Dipl.-Inform. J. Hoefinghoff, J. Kapitza, M. Cherubim

Übungsblatt 5

Seite 3 von 3

```
...  
  
Order eingegangen:  
GrafikkarteA    10      2500.0 EUR  
ProzessorB      2       720.0 EUR  
  
Orders gesamt  
=====
```

ProzessorA	1	400.0 EUR
GrafikkarteB	3	360.0 EUR
GrafikkarteA	12	3000.0 EUR
ProzessorB	2	720.0 EUR

```
=====
```

Gesamtanzahl:	18
Gesamtwert:	4480.0 EUR

```
=====
```

...

Abbildung 1: Versandlagerausgabe

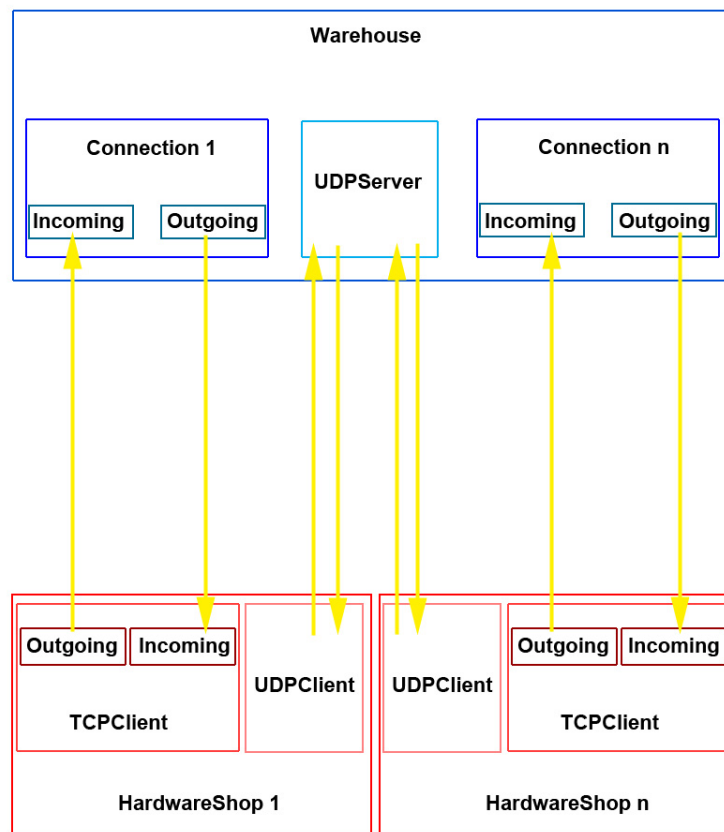


Abbildung 2: Übersicht