

## Übungen zu Informatik B

*Sommersemester 2004*

### Blatt 10

**Dies ist das letzte Aufgabenblatt.**

*Die Klausur zu Informatik B findet am Montag, den 12.7.2004, um **Punkt 8:00 Uhr (s.t.!) im Raum 22/B01 statt (Innenstadt, Heger-Tor-Wall 14).***

#### **Aufgabe 1 (6 Punkte)**

Erläutern Sie Ihrer/m Tutor(in) die folgenden Begriffe:

- *Protokoll*
- TCP und UDP
- Internet-Adresse, Port
- Socket, ServerSocket

#### **Aufgabe 2 (6 Punkte)**

In der Literatur gibt es ein klassisches Synchronisationsproblem, das unter dem Namen *Lamport's Bakery* bekannt ist:

In einer Bäckerei verkaufen  $n$  Verkäufer Waren an Kunden. Jeder Kunde, der den Laden betritt, zieht aus einem Automaten eine Nummer. Wenn ein Verkäufer mit der Bedienung eines Kunden fertig ist, ist als nächstes derjenige Kunde an der Reihe, der die kleinste Nummer in der Hand hält und noch nicht von einem anderen Verkäufer bedient wird.

Verwenden Sie diese Idee, um eine „faire“ Behandlung der Konsumenten in dem in der Vorlesung gezeigten Produzent/Konsument Beispiel zu erreichen. Als Grundlage soll eine Lösung des Produzent/Konsument Problems mit mehreren Konsumenten, aber nur *einem* Produzenten, dienen (`code/21/ManyProducerManyConsumerLimitedWithoutDeadlock.java`).

Die Konsumenten ziehen nun also vor jedem Schritt eine Nummer, und dürfen dann in aufsteigender Reihenfolge ihrer Nummern Daten aus dem „Lager“ entnehmen (natürlich nur, wenn auch Daten vorhanden sind).

Die Vergabe der Nummern soll durch ein Objekt einer eigenen Klasse geschehen, das auf Anfrage jedem Konsumenten eine (fortlaufend gezählte) Nummer zuteilt und darüber buchführt, welcher Konsument (d.h. welche Nummer) gerade an der Reihe ist.

— bitte wenden —

### Aufgabe 3 (8 Punkte)

Implementieren Sie einen Server für einen einfachen „Telefonbuch-Dienst“:

Ihr Server registriert sich auf einem beliebigen *Port* und wartet dort auf eingehende Verbindungen. Wenn ein Client eine Verbindung zu diesem Port aufgebaut hat, schickt zunächst der Client eine Zeile mit dem Namen, zu der die Telefonnummer abgefragt werden soll. Der Server antwortet darauf mit einer Textzeile, die die entsprechende Telefonnummer oder den Text `UNKNOWN` enthält. Alle übertragenen Zeilen werden jeweils durch die Sequenz `"\r\n"` als Zeilentrenner abgeschlossen. Ein Client kann mehrere Anfragen hintereinander stellen, jeweils auf separaten Zeilen.

Das „Telefonbuch“ wird im Server durch eine (feste) *HashMap* repräsentiert, die Paare von Namen und zugeordneten Telefonnummern enthält. Beim Start des Servers wird diese Collection mit frei wählbaren Beispieldaten initialisiert (als freiwillige Erweiterung kann der Inhalt auch aus einer Datei gelesen werden).

Der Server soll in der Lage sein, mehr als einen Client gleichzeitig zu bedienen. Dazu wird vom Server nach dem Akzeptieren einer Verbindung ein eigener Thread gestartet, der sich um die Kommunikation mit diesem Client kümmert. Beendet der Client die Verbindung, geht auch der zugehörige Thread im Server zu Ende. Der Haupt-Thread des Servers kümmert sich also nur noch um die Einrichtung des Ports, auf dem die Verbindungen eingehen, und die Annahme von neuen Verbindungen.

Die vom Server verwendete Port-Nummer kann über ein Kommandozeilenargument beim Aufruf des Programms gesetzt werden.

Sie können Ihren Telefonbuch-Server mit dem *Echo-Client* aus der Vorlesung (wobei der Port geändert werden muß) oder dem `telnet`-Kommando testen.

**Gesamtpunkte: 20**