

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	<b>HERBST</b> <b>1993</b>	<b>66111</b>
Kennwort: _____		
Arbeitsplatz-Nr.: _____		

**Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen**  
**- Prüfungsaufgaben -**

**Fach:** Informatik (vertieft studiert)  
**Einzelprüfung:** Betriebs/Datenbanksyst., Rechn.architekt  
**Anzahl der gestellten Themen (Aufgaben):** 1  
**Anzahl der Druckseiten dieser Vorlage:** 4

**Sämtliche Teilaufgaben sind zu bearbeiten!**

**Aufgabe 1:**

Betrachten Sie für die folgenden Teilaufgaben eine von einem Polizisten geregelte Straßenkreuzung, für die gelten soll:

- Es kreuzen sich zwei Einbahnstraßen. Die Autos wollen die Kreuzung in Nord-Süd Richtung und Ost-West Richtung überqueren (Abbiegen ist nicht möglich).
- Es kann sich immer nur ein Auto in der Kreuzung befinden.  
Vor der Kreuzung kann sich ein Stau bilden (nach der Kreuzung nicht).
- Der Polizist regelt den Verkehr fair und möglichst optimal. Er kann die Durchfahrt (wie an Kreuzungen üblich) in Nord-Süd oder Ost-West Richtung freigeben und die andere Richtung jeweils sperren.  
Er wechselt die Richtung spätestens nach 20 sec, schaltet aber dann schon eher um, wenn in der freien Richtung keine Autos mehr kommen, in der gesperrten Richtung aber welche warten.

**Fortsetzung nächste Seite!**

- Beim Umschalten der Richtung hebt er zuerst den Arm, worauf keiner mehr in die Kreuzung einfahren darf. Erst wenn sich kein Auto mehr in der Kreuzung befindet, schaltet er um.

Bemerkung für die folgenden Teilaufgaben: Man kann nicht damit rechnen, daß sich Autofahrer immer an die Verkehrsregeln halten. Aus Sicherheitsgründen sollte deshalb das Modul "Straßenkreuzung" (soweit möglich) die Operationenfolge der durchfahrenden Autos an der Kreuzung festlegen.

### Aufgabe 1.1:

Entwerfen Sie eine formale Spezifikation eines Moduls "Straßenkreuzung" für das beschriebene Problem! Das Modul soll dabei die folgenden Operationen und Prozesse beinhalten.

a) die Operationen des Moduls Kreuzung:

*an\_kreuzung\_heranhfahren* (auto: wagen, richt: richtung)

*in\_kreuzung\_einfahren* (richt: richtung)

*aus\_kreuzung\_herausfahren* ( )

*über\_kreuzung\_fahren* (auto: wagen, richt: richtung) /\* diese Operation ruft die drei anderen Operationen in der richtigen Reihenfolge auf \*/

b) den Prozeß

*polizist*

c) Zusätzlich enthält das Modul die Operation Zeittakt, die jede Sekunde von einem Uhrmodul aufgerufen wird.

*zeit\_takt* ( )

EFFECTS: zeit = 'zeit + 1

Anmerkung zur Notation: In Prädikaten, die den Effekt einer Operation beschreiben, wird der Variablenname als Bezeichner des Wertes verwendet, den die Variable nach Ausführung der Operation hat, und der Variablenname mit vorgestelltem Apostroph für den Wert unmittelbar vor Ausführung der Operation.

d) Es seien die folgenden Typen bekannt:

TYPES richtung = (ost\_west, nord\_süd)  
 wagen = autonummer;  
 queue /\* oder sequence, mit den üblichen Operationen  
 first, tail und add, remove \*/

END\_TYPES

**Aufgabe 1.2:**

Implementieren Sie für das eingangs beschriebene Problem ein Modul "Straßenkreuzung" mit den in Aufgabe 1.1 genannten Operationen und Prozessen! Ihre Implementierung soll insbesondere die auftretenden Synchronisationsprobleme lösen.

Sie können selbst festlegen, welches Synchronisationsmittel Sie verwenden wollen (z.B. Semaphore mit den Funktionen  $p$  und  $v$ ). Begründen Sie Ihre Wahl!

(Sie können nach Bedarf auch zusätzliche Hilfsfunktionen und/oder Prozesse einführen.)

**Aufgabe 2: (Warteschlangen)**

In der Mensa existiere nur eine Kasse. Da sich an der einen Kasse regelmäßig längere Wartezeiten ergeben, werden zur Beschleunigung der Abfertigung die 3 folgenden Lösungen diskutiert:

- a) Es wird eine zweite Kasse installiert. Die Studenten stellen sich in einer Warteschlange an und der jeweils erste wartende Student benutzt die erste freiwerdende Kasse.
- b) Durch organisatorische Maßnahmen kann erreicht werden, daß die Abfertigung an der einen Kasse doppelt so schnell erfolgt.
- c) Es werden zwei getrennte Warteschlangen mit zwei Kassen mit der ursprünglichen Abfertigungsgeschwindigkeit installiert. Die Studenten verteilen sich gleichmäßig auf die beiden Warteschlangen.

Sowohl die Ankunftsintervalle der Studenten (Ankunftsrate  $\lambda$ ) als auch die Bedienzeiten an der Kasse (Bedienrate  $\mu$ ) sollen exponentiell verteilt sein.

**Aufgabe 2.1:**

Entwerfen Sie zu jedem der gemachten Vorschläge ein Warteschlangenmodell!

**Aufgabe 2.2:**

Bestimmen Sie für die einzelnen Vorschläge die mittlere Verweilzeit im System!

Welches ist somit das beste System (mit der kürzesten Verweilzeit im System)?

Hinweis: Für ein System aus  $k$  Bedienstationen mit exponentiell verteilter Ankunfts- und Bedienrate ( $M/M/k$ ) gilt für den Erwartungswert der Anzahl von Aufträgen im System:

$$E[N] = k\rho + p_0 \frac{(\rho(k\rho)^k)}{k!(1-\rho)^2} \quad \text{mit } p_0 = \frac{1}{\frac{(k\rho)^k}{k!(1-\rho)} + \sum_{i=0}^{k-1} \frac{(k\rho)^i}{i!}} \quad \text{und } \rho = \frac{\lambda}{k\mu}$$

und es gilt der Satz von Little:

mittlere Verweilzeit =  $E[N]$  / mittlere Anzahl ankommender Aufträge pro Zeiteinheit.

**Aufgabe 2.3:**

Wie verhalten sich die 3 Vorschläge bezüglich der Ausfallsicherheit?