

Technische Universität Berlin  
Institut für Softwaretechnik und Theoretische Informatik  
FG Softwaretechnik  
Franklinstr. 28/29  
10587 Berlin

Helke  
Mertgen

## Beispielklausur B

### MPGI 3

Prüfen Sie zunächst, ob dieses Exemplar vollständig ist (6 beidseitig bedruckte Blätter).

Tragen Sie auf diesem Titelblatt und darüber hinaus auf allen Blättern, die Sie für Ihre Niederschrift verwenden, Ihren Namen und Ihre Matrikelnummer ein. Zum Bestehen der Klausur sind mindestens **25 Punkte** notwendig. Insgesamt sind 50 Punkte möglich. Die Bearbeitungszeit beträgt 75 Minuten. In der Klausur sind außer einem beschriebenen DIN-A4-Blatt **keine Hilfsmittel** zugelassen. Verwenden Sie ausschließlich das ausgeteilte Klausur-Papier. Es dürfen nur **permanent-schwarze oder blaue Stifte** zum Lösen der Aufgaben verwendet werden.

Viel Erfolg!

Name, Vorname: \_\_\_\_\_

Matrikelnummer: \_\_\_\_\_

Studienrichtung: \_\_\_\_\_

MPGI3-Übungen habe ich im Semester \_\_\_\_\_ besucht.

Aufgabe	1a	1b	1c	2	Gesamt
Maximal:	6	10	14	20	50
Erreicht:					

## Aufgabe 1 (Object-Z) - Verladestation

An einem Flughafen gibt es eine Verladestelle zum Ein- und Ausladen von Koffern. Ein Flugzeug, das zu der Verladestelle kommt, wird zunächst vollständig entladen. Danach wird es mit Koffern beladen, die das selbe Ziel haben, wie das Flugzeug auf seinem kommenden Flug.

### Aufgabenstellung:

Spezifizieren Sie die Verladeaktionen in Object-Z, indem Sie die globalen Definitionen und die vorgegebenen Klassendefinitionen ergänzen.

Globale Definitionen:

[FLUGHAFEN]  
Nachricht ::=

<p><i>Koffer</i> _____   ziel : FLUGHAFEN</p>
---

### 1a) Flugzeug

6 Punkte

Ein Flugzeug, das zu der Verladestelle kommt, hat als aktuelles Flugziel den Flughafen, den es als nächstes ansteuert. Jedes Flugzeug hat eine Kapazität, welche die maximale Anzahl von Koffern festlegt, die es transportieren kann. Die Menge der beförderten Koffer darf niemals die Kapazität übersteigen. Das Flugzeug gilt als vollbeladen, wenn die Kapazität ausgeschöpft ist.

Initial transportiert ein Flugzeug keine Koffer.

Die Operation *Ausladen* entlädt sämtliche Koffer aus dem Flugzeug und gibt die Menge dieser Koffer zurück.

Die Operation *Einladen* lädt einen Koffer in das Flugzeug und bestätigt die erfolgreiche Aktion mit einer Nachricht. Der Koffer wird nur unter der Vorbedingung angenommen,

- dass das Flugzeug noch nicht vollbeladen ist
- und dass das Ziel des Koffers mit dem Flugziel übereinstimmt.

Flugzeug

↑ (INIT, Einladen, Ausladen, ziel, vollbeladen)

| kapazität :  $\mathbb{N}$

ziel : FLUGHAFEN

koffer :  $\mathbb{P}$  Koffer

vollbeladen :  $\mathbb{B}$

INIT

Ausladen

$\Delta$ (koffer, vollbeladen)

ks! :  $\mathbb{P}$  Koffer

Einladen

$\Delta$ (koffer, vollbeladen)

k? : Koffer

n! : Nachricht

## 1b) Verladestelle

10 Punkte

An der Verladestelle wartet eine Menge von Flugzeugen, die beladen werden sollen. Es gibt ein Lager mit allen eingegangenen Koffern und ein Lager mit allen zum Verladen aufgegebenen Koffern. Die Menge der aufgegebenen Koffer ist nach der Reihenfolge, in der sie aufgegeben wurden, nummeriert. Es gilt, dass keiner der aufgegebenen Koffer im Lager mit den eingegangenen Koffern liegen darf und umgedreht.

Initial gibt es weder Koffer noch Flugzeuge in der Verladestelle.

### Flugzeug an der Verladestation annehmen

Die Operation *Flugzeug\_hinzufügen* fügt der Menge der wartenden Flugzeuge einen weiteren Flugzeug hinzu.

Die Operation *Einlagern* erwartet eine Menge von Koffern als Eingabe und fügt diese der Menge der eingegangenen Koffer hinzu. (Die eingegangenen Koffer können dann von den Fluggästen abgeholt werden, was aber nicht mehr Bestandteil dieser Spezifikation ist.)

Die robuste Operation *Flugzeug\_annehmen* beschreibt die Annahme eines ankommenden Flugzeugs bei der Verladestation. Die Operation erwartet ein Flugzeug als Eingabe, für das der folgende Ablauf spezifiziert werden soll:

- Alle Koffer des Flugzeugs werden ausgeladen.
- Die ausgeladenen Koffer werden in der Verladestation eingelagert.
- Das Flugzeug wird der Menge der wartenden Flugzeuge hinzugefügt.

Verladestelle

↓ (INIT, *Flugzeug\_annehmen*, *Koffer\_aufgeben*)

*flugzeuge* :  $\mathbb{P}$  *Flugzeug*  
*aufgegeben* :  $\mathbb{N} \leftrightarrow$  *Koffer*  
*eingegangen* :  $\mathbb{P}$  *Koffer*

INIT

*Flugzeug\_hinzufügen*

$\Delta($  \_\_\_\_\_  $)$   
*f?* : *Flugzeug*

*Einlagern*

$\Delta($  \_\_\_\_\_  $)$

*Flugzeug\_annehmen*  $\hat{=}$  [*f?* : *Flugzeug*] •

### 1c) Verladestelle (Fortsetzung)

14 Punkte

#### Koffer aufgeben und verladen

Mit der Operation *Koffer\_aufgeben* kann ein Fluggast einen Koffer zum Verladen aufgeben. Die Koffer sollen in der Reihenfolge ihrer Aufgabe vermerkt werden, um sicher zu stellen, dass kein Koffer übermäßig lange auf seinen Transport warten muss.

Die Operation *Nächsten\_Koffer\_auslagern* liefert den aufgegebenen Koffer, der bis jetzt am längsten im Lager auf seinen Transport gewartet hat. Der Koffer wird aus dem Lager entfernt.

Mit der Operation *Verladen\_nicht\_möglich* wird spezifiziert, unter welchen Bedingungen das Verladen eines bestimmten Koffers scheitern kann. Sie soll berücksichtigen, ob derzeit ein Flugzeug an der Verladestation wartet, das diesen Koffer annehmen würde.

Die Operation *Koffer\_zurückstellen* erwartet einen Koffer als Eingabe. Sie spezifiziert den Fall, dass das Verladen dieses Koffers derzeit nicht möglich ist und der Koffer daher erneut aufgegeben wird.

Mit der Operation *Koffer\_verladen* soll schließlich der gesamte Verladevorgang für einen Koffer spezifiziert werden:

- Der jeweils nächste Koffer aus dem Lager wird ausgelagert,
- falls möglich, wird aus der Menge der wartenden Flugzeuge eines ausgewählt, in das der Koffer verladen wird,
- andernfalls wird der Koffer zurückgestellt.

**Hinweis:** Sie können die Funktionen **min** und **max** verwenden, die das minimale bzw. das maximale Element einer Menge von natürlichen Zahlen ergibt.

Verladestelle (Fortsetzung)

Koffer\_aufgeben

$\Delta($  )  
 $k? : \text{Koffer}$

Nächsten\_Koffer\_auslagern

$\Delta($  )

Verladen\_nicht\_möglich

$k? : \text{Koffer}$   
 $n! : \text{Nachricht}$

Koffer\_zurückstellen  $\hat{=}$

Koffer\_verladen  $\hat{=}$

Beispielklausur B MPCI3, Name: \_\_\_\_\_ MatrNr.: \_\_\_\_\_ 7

## Aufgabe 2 (Statecharts) - Spülmaschine

Es soll eine (Geschirr-)Spülmaschine mit Hilfe von Statecharts modelliert werden. Die Spülmaschine besteht aus den zu modellierenden Komponenten *Tür*, einem Knopf für das *Intensiv*-Programm, einem Knopf zum *Vorspülen* und einer Steuerung für den *Programmablauf*. Die Spülmaschine ist ein Haushaltsgerät zum Reinigen von Geschirr, das über die Tür ein- und ausgeräumt werden kann.

Der Programmablauf kann mit der *Start-Taste* gestartet und mit der *Stop-Taste* unterbrochen werden. Wird direkt nach einer Unterbrechung die *Start-Taste* erneut gedrückt, so wird der Programmablauf mit dem zuletzt aktiven Programmpunkt fortgesetzt. Der zuletzt aktive Programmpunkt wird hierbei noch einmal komplett (von Anfang bis Ende) ausgeführt. Wird hingegen direkt nach einer Unterbrechung noch einmal die *Stop-Taste* gedrückt, so beendet die Spülmaschine das Programm endgültig und schaltet sich aus.

Ein Programm enthält folgende Programmpunkte und läuft wie folgt ab.

1. *Vorspülen* (optional, 15 Minuten)
2. *Reinigen* (1 Stunde)
3. *Klaraspülen* (30 Minuten)
4. *Trocknen* (30 Minuten)

Der Programmablauf innerhalb der einzelnen Programmpunkte wird durch externe Module (Teil der Umgebung) realisiert und ist bis auf die angegebenen Zeitangaben nicht genauer zu modellieren.

Der Programmablauf lässt sich nur starten, wenn die Tür geschlossen ist. Der Programmpunkt *Vorspülen* kann mit dem entsprechenden Knopf an- oder abgeschaltet werden. Mit dem Knopf für das *Intensiv*-Programm wird gesteuert, ob der Programmpunkt *Reinigen* einmal oder zweimal ausgeführt werden soll. Die Einstellungen zum *Vorspülen* und *Reinigen* können nach dem Programmstart nicht mehr verändert werden. Nach dem Trocknen ist das Programm beendet und die Maschine schaltet sich aus. Während eines Programmablaufs lässt sich die Tür nicht öffnen. Nur im ausgeschalteten Zustand kann die Tür geöffnet werden.

Initial ist die Spülmaschine ausgeschaltet, die Tür geschlossen und die Programm-Optionen (*Vorspülen*, *Intensiv*) sind deaktiviert.

## Aufgabenstellung

20 Punkte

Modellieren Sie die beschriebene Spülmaschine mit ihren Komponenten und dem Programmablauf mit Hilfe von Statecharts. Es sollen nur die beschriebenen Eigenschaften modelliert werden. (Das Ein- / Ablaufen von Wasser etc. soll z.B. **nicht** modelliert werden.)

Verwenden Sie die in der folgenden Tabelle aufgeführten Ereignisse, die durch den Benutzer ausgelöst werden können. Wenn Sie neue Ereignisse für die Modellierung definieren wollen, ergänzen Sie entsprechend die Tabelle.

NAME	BESCHREIBUNG
EXTERNE EINGABEREIGNISSE	
<b>start</b> <b>stop</b> <b>auf</b> <b>zu</b> <b>i_ein</b> <b>i_aus</b> <b>v_ein</b> <b>v_aus</b>	Die Start-Taste der Spülmaschine wird gedrückt. Die Stop-Taste der Spülmaschine wird gedrückt. Die Taste zum Tür-Öffnen wird gedrückt. Die Taste zum Tür-Schließen wird gedrückt. Der Knopf für das Intensiv-Programm wird aktiviert. Der Knopf für das Intensiv-Programm wird deaktiviert. Der Knopf für das Vorspülen wird aktiviert. Der Knopf für das Vorspülen wird deaktiviert.
EXTERNE AUSGABEREIGNISSE	
INTERNE EREIGNISSE	

Beispielklausur B MPCI3, Name: \_\_\_\_\_ MatrNr.: \_\_\_\_\_ 10

**Statechart-Lösung:**