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

Jähnichen
Bittner
Helke
Seifert

Klausur

Softwaretechnik

23.7.2003

Prüfen Sie zunächst, ob dieses Exemplar vollständig ist (12 Blätter). Es dürfen keine eigenen Blätter verwendet werden.

Tragen Sie auf diesem Deckblatt und darüber hinaus auf allen Blättern, die Sie für Ihre Niederschrift verwenden, Ihren Namen und Ihre Matrikelnummer ein.

Zum Bestehen der Klausur sind **35 Punkte** notwendig. Insgesamt sind 100 Punkte möglich. Die Bearbeitungszeit beträgt 120 Minuten. In der Klausur sind **außer Telekommunikationseinrichtungen sämtliche Hilfsmittel** zugelassen.

Viel Erfolg!

Name, Vorname: ______________________________

Matrikelnummer: ______________

Studienrichtung/Prüfungsordnung: ______________
SWT-Übungen habe ich im Semester ________besucht.

Aufgabe	1	2	3	Gesamt
Maximal:	40	40	20	100
Erreicht:				

Aufgabe 1 (Fusion) - Squashcenter

(Geschätzte Bearbeitungszeit: 45 min.) Ein Squashcenter beauftragt eine Firma mit der Entwicklung eines Softwaresystems, das das Personal bei der Verwaltung der ablaufenden Geschäftsprozesse unterstützen soll.

Im Eingangsbereich des Squashcenters befindet sich eine Theke, an der Spieler mit dem Personal telefonisch oder persönlich Kontakt aufnehmen können, um z.B. Reservierungen vorzunehmen.

Das Squashcenter ist täglich von 8 bis 23 Uhr geöffnet und verfügt über 10 Plätze. Ein Zeitslot für einen Platz ist 45 Minuten lang. Somit stehen 20 Zeitslots für jeden Platz pro Tag zur Verfügung.

Reserviert ein Spieler einen Platz, so wird für diesen Spieler der Zeitslot und der Reservierungstag vermerkt. Gleichzeitig wird vermerkt, dass der entsprechende Zeitslot an diesem Tag reserviert ist. Ein Spieler kann maximal zwei Reservierungen vornehmen lassen.

Weiterhin kann er an der Theke Zubehör ausleihen. Zubehör kann ein Ball, ein Schläger oder ein Paar Squashschuhe sein. Ein Spieler kann maximal fünf Zubehörteile ausleihen.

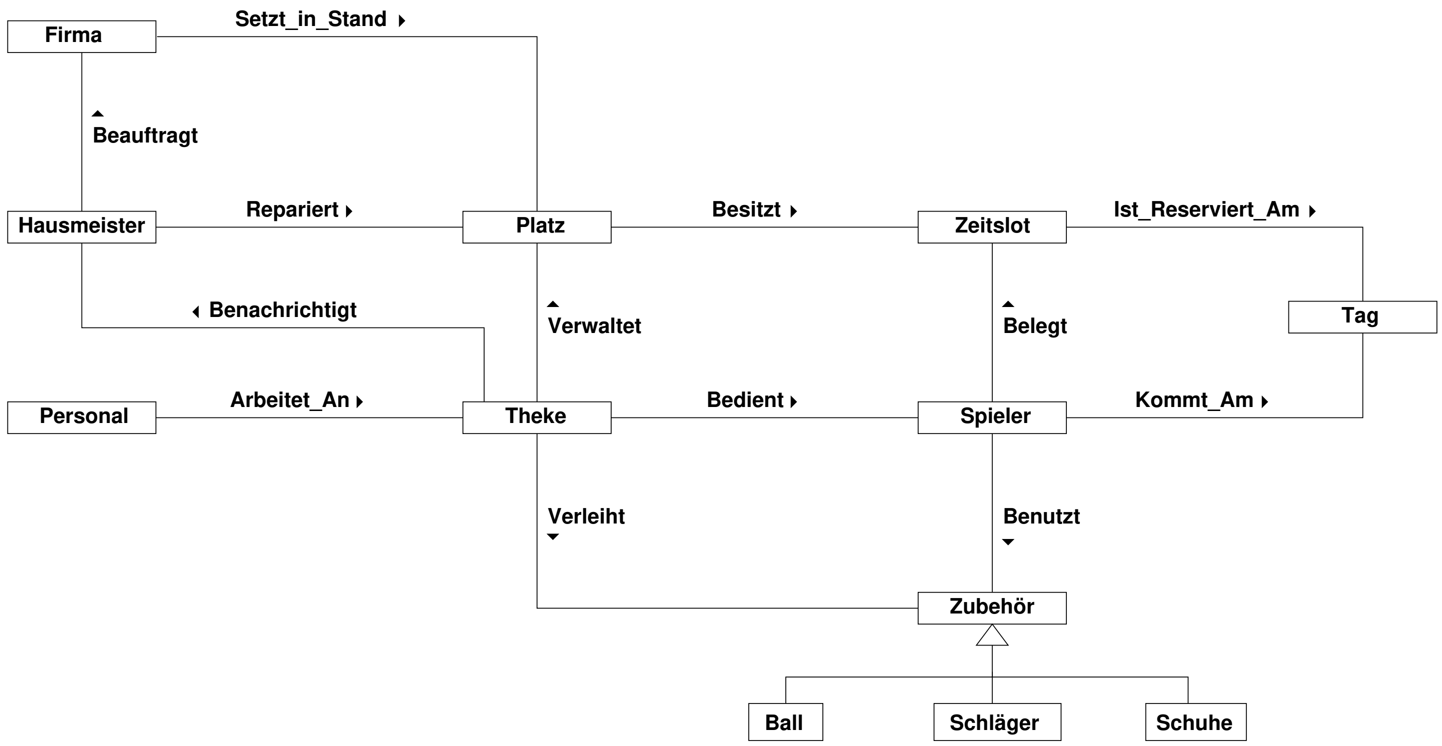
Zusätzlich kann das Personal beschädigte Plätze sperren und einen der beiden Hausmeister über das System mit der Reparatur beauftragen. Dabei soll vermerkt werden, welcher der Hausmeister mit der Reparatur beauftragt ist. Der Hausmeister überprüft dann den Schaden und entscheidet, ob er den Schaden selbst beheben kann oder ob er eine Firma mit der Reparatur beauftragen muß. Nach erfolgter Reparatur vermerkt er, dass der Platz wieder bespielt werden kann.

Aufgabenstellung

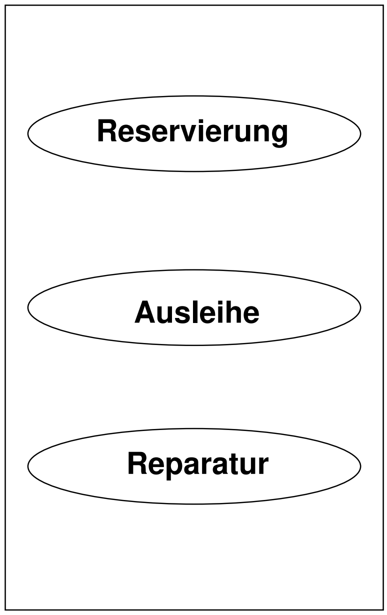
Das System soll nach der Fusion-Methode entwickelt werden.

1. Ergänzen Sie im gegebenen Klassenmodell die Multiplizitäten. *3 Punkte*
2. Vervollständigen Sie das Use-Case-Modell, indem Sie die Akteure des Systems bestimmen und ihnen Funktionsgruppen zuordnen. *3 Punkte*
3. Vervollständigen Sie das vorgegebene Systemklassenmodell, indem Sie die Systemgrenze festlegen und gegebenenfalls weitere neue Klassen und dazugehörige Assoziationen einfügen. Multiplizitäten müssen nicht angegeben werden. *8 Punkte*
4. Vervollständigen Sie das Operationsschema für die Systemoperation *buchen*. *15 Punkte*
5. Geben Sie den Objektinteraktionsgraphen für die Systemoperation *schaden_melden* an. Verwenden Sie das Konzept der Delegation. *11 Punkte*

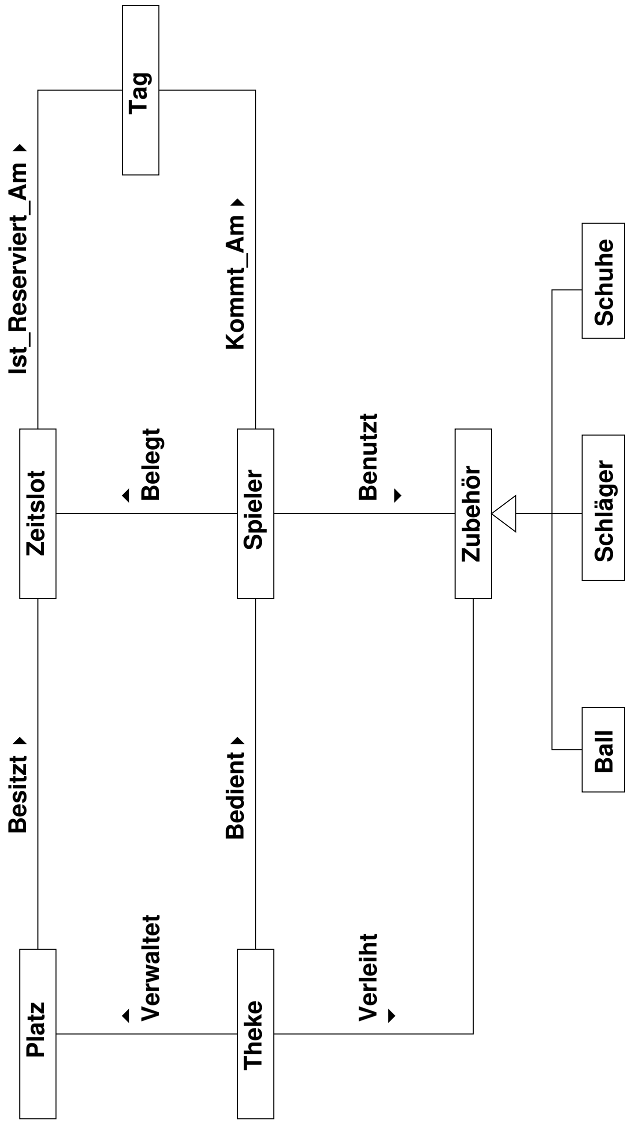
Klassenmodell



UseCase-Modell



Systemklassenmodell

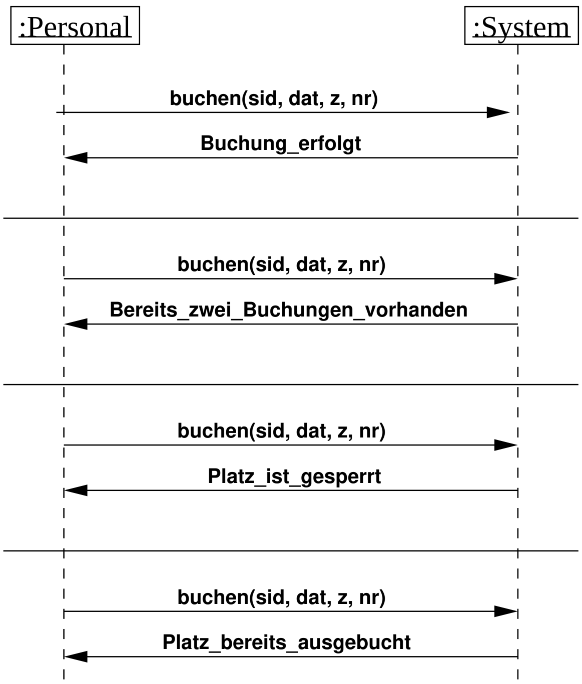


Operationsmodell

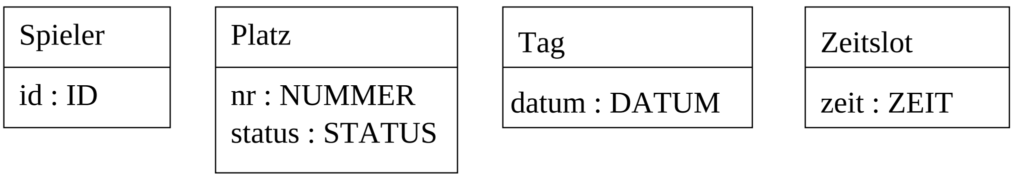
Bei der Spezifikation der Systemoperation *buchen* sollen Sie folgende Vorbedingungen annehmen:

1. Die Daten des Spielers, der die Buchung vornehmen möchte, sind dem System bekannt.
2. Die Nummer des Platzes, der gebucht werden soll, ist eine gültige Platznummer.
3. Alle Objekte der Klasse *Tag* sind dem System bereits bekannt.

Die Eingaben (Id des Spieler, Datum, Zeit und Nummer des Platzes) und die zu behandelnden Fehlerfälle von *buchen* sind durch folgendes Timeline-Diagramm gegeben:



Folgende Attribute und Typen können in der Spezifikation des Operationsschemas verwendet werden:




```
STATUS = gesperrt | bespielbar
```

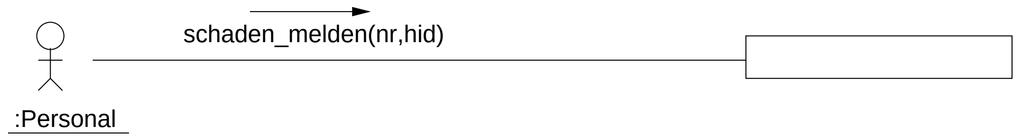
Op.	=	buchen
Desc.	=	Ein Spieler bucht einen Platz für ein Zeitslot und Tag. Ist der Platz defekt oder steht kein Zeitslot mehr zur Verfügung oder hat der Spieler bereits zwei Buchungen vornehmen lassen, so wird eine entsprechende Fehlermeldung ausgegeben.
Input	=	$sid : ID;\ dat : DATUM;\ z : ZEIT;\ nr : NUMMER$
Reads	=	
Changes	=	
Sends	=	
Pre	=	—
Post	=	

Objektinteraktionsmodell

Gegeben sei folgendes Operationsschema.

Op.	=	schaden_melden
Desc.	=	Ein Mitarbeiter möchte einen beschädigten Platz sperren. Sollte dieser Platz bereits als beschädigt markiert sein, so wird eine Fehlermeldung ausgegeben. Ist der Platz noch nicht als beschädigt gemeldet, so wird ein Hausmeister mit der Reparatur beauftragt und die Schadensmeldung bestätigt. *Hinweis:* In der Assoziation *Repariert* wird gespeichert, welcher Hausmeister mit der Reparatur beauftragt wurde.
Input	=	$nr : NUMMER, hid : HAUSMEISTERID$
Reads	=	$h : HM\text{-}Daten\ with\ h.id = hid$ $Benachrichtigt$
Changes	=	$p : Platz\ with\ p.nr = nr$ $Repariert$
Sends	=	$\underline{: Personal}\ \{Schaden\_aufgenommen, Schaden\_bereits\_gemeldet\}$ $\underline{: Hausmeister}\ \{Neuer\_Schaden\_gemeldet(nr)\}$
Pre	=	$-$
Post	=	$(p.status = gesperrt \Rightarrow$ $\quad \textbf{is\_sent}\ \{Schaden\_bereits\_gemeldet\} \wedge \textbf{no\_effect}\ ) \wedge$ $(p.status = bespielbar \Rightarrow$ $\quad p.status' = gesperrt \wedge$ $\quad Repariert' = Repariert \cup \{(h, p)\} \wedge$ $\quad \textbf{is\_sent}\ \{Neuer\_Schaden\_gemeldet(nr)\} \wedge$ $\quad \textbf{is\_sent}\ \{Schaden\_aufgenommen\})$

Entwickeln Sie aus dem gegebenen Operationsschema den dazugehörigen Objektinteraktionsgraphen.



Aufgabe 2 (Object-Z) - Flugüberwachungssystem

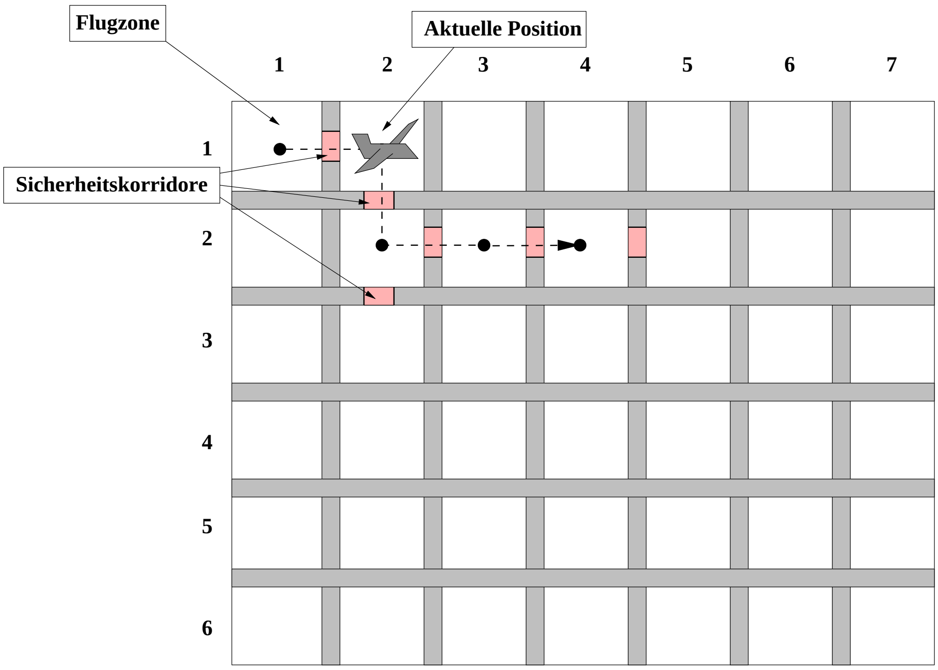
(Geschätzte Bearbeitungszeit: 45 min.)

Es soll zur Unterstützung von Fluglotsen ein Flugüberwachungssystem spezifiziert werden, das einen reibungslosen Flugverkehr von Flugzeugen innerhalb eines Flugraumes unterstützt.

Der Flugraum ist durch ein Raster in einzelne Flugzonen unterteilt. Die Rasterstruktur sei durch ein zweidimensionales Koordinatensystem beschrieben.

Flugzeuge, die zwischen Flugzonen wechseln wollen, können dies nur über Sicherheitskorridore tun. Ein Sicherheitskorridor ist gerichtet und kann immer nur ein Flugzeug zur gleichen Zeit beinhalten.

Die folgende Abbildung zeigt die schematische Veranschaulichung des Luftraumes.



Aufgabenstellung

Spezifizieren Sie das beschriebene Flugüberwachungssystem in Object-Z, indem Sie die vorgegebenen Klassendefinitionen ergänzen. Verwenden Sie dabei die gegebenen Typen:

$$\begin{aligned}
&Koordinate == \mathbb{N}_1 \times \mathbb{N}_1 \\
&Nachricht ::= Flugzone_betreten \mid Flugzone_voll \mid \\
&\qquad\qquad Flugzone_verlassen \mid Flugzone_leer
\end{aligned}$$

Flugzeug

5 Punkte

Flugzeuge werden durch die Klasse *Flugzeug* beschrieben. Ein Flugzeug ist durch seine aktuelle *Position* im Koordinatensystem des Luftraumes charakterisiert.

Ein Flugzeug kann durch die Operation *Bewegen* seine Position verändern. Dabei ist zu beachten, dass nur direkt benachbarte Koordinaten als neues Ziel angegeben werden dürfen. Diagonale Überschreitungen sind ausgenommen.

Flugzeug

$position : Koordinate$

Bewegen

$start! : Koordinate$
$ziel! : Koordinate$

Flugzone

12 Punkte

Eine Flugzone ist durch eine Koordinate im Flugraum charakterisiert und verwaltet eine endliche Anzahl von Flugzeugen. Die Anzahl der in einer Flugzone befindlichen Flugzeuge darf eine konstante Zahl *max* nicht überschreiten. Die aktuelle Position von allen verwalteten Flugzeugen muß mit der Koordinate der Flugzone übereinstimmen. Initial sind in einer Flugzone keine Flugzeuge enthalten.

Die Operation *Eintreten* ist aus einer Operation für den Normalfall (*EintretenOK*) und einer Operation für den Fehlerfall (*FlugzoneVoll*) zusammengesetzt. Sie beschreibt das Eintreten eines Flugzeuges in eine Flugzone. Die Operation *Eintreten* besitzt die Vorbedingung, dass ein eintretendes Flugzeug noch nicht in der Flugzone vorhanden ist.

Die Operation *Austreten* ist aus einer Operationen für den Normalfall (*AustretenOK*) und einer Operation für den Fehlerfall (*FlugzoneLeer*) zusammengesetzt. Sie beschreibt das Austreten eines Flugzeuges aus der Flugzone.

Flugzone

$max : \mathbb{N}$

INIT

EintretenOk

$f? : Flugzeug$
$ne! : Nachricht$

FlugzoneVoll

$f? : Flugzeug$
$ne! : Nachricht$

AustretenOK

$f! : Flugzeug$
$nv! : Nachricht$

FlugzoneLeer

$f! : Flugzeug$
$nv! : Nachricht$

$Eintreten \mathrel{\widehat=}$

$Austreten \mathrel{\widehat=}$

Sicherheitskorridor

10 Punkte

Ein *Sicherheitskorridor* befindet sich zwischen benachbarten Flugzonen und ist durch eine Start- und eine Zielkoordinate charakterisiert. Weiterhin ist bekannt, welches Flugzeug sich aktuell im Sicherheitskorridor befindet. Es darf immer nur ein Flugzeug den Sicherheitskorridor passieren. Initial befindet sich kein Flugzeug im Sicherheitskorridor.

Mit den Operationen *Freigeben* und *Sperren* kann ein Sicherheitskorridor für die Benutzung freigegeben oder gesperrt werden.

Mit der Operation *Betreten* kann ein übergebenes Flugzeug den Sicherheitskorridor betreten. Die Vorbedingung dieser Operation ist, dass der Sicherheitskorridor bereits gesperrt ist. Ausserdem werden in den Ausgabeparametern die Start- und Zielkoordinaten des Sicherheitskorridors zurückgegeben.

Mit der Operation *Besetzen* soll für ein übergebenes Flugzeug der Sicherheitskorridor gesperrt, eine Positionsveränderung im Flugzeug über die Operation *Bewegen* herbeigeführt und der Sicherheitskorridor betreten werden. Dabei wird sichergestellt, dass die Start- und Zielkoordinaten des Flugzeuges mit denen des Sicherheitskorridors übereinstimmen.

Sicherheitskorridor

INIT

Sperren

Freigeben

Betreten

$\Delta(flugzeug)$
$f? : Flugzeug$
$start! : Koordinate$
$ziel! : Koordinate$

$Besetzen \mathrel{\widehat{=}}$

Flugraum

13 Punkte

In einem *Flugraum* werden die *Flugzonen* in einer Funktion verwaltet, die Koordinaten auf Flugzonen abbildet. Die *Sicherheitskorridore* zwischen den Flugzonen werden in einer Menge abgelegt. Anfragende Flugzeuge, die in eine benachbarte Flugzone wechseln wollen, werden ebenfalls in einer Menge aufgesammelt. Diese Menge enthält nur Flugzeuge, die in Flugzonen des Luftraumes vorkommen.

Durch die Operationen *AnfrageAufnehmen* wird das Hinzufügen und durch die Operation *AnfrageEntnehmen* wird das Entfernen von anfragenden Flugzeugen realisiert.

Mit der Operation *Wechseln* wird für eine übergebene Start- und Zielkoordinate der Wechsel von einem Flugzeug zwischen zwei Flugzonen ohne Verwendung eines Sicherheitskorridors beschrieben.

In der Operation *AbgesichertWechseln* wird die Operation *Wechseln* wiederverwendet. Zunächst wird nichtdeterministisch ein Sicherheitskorridor ausgewählt. Der gewählte Sicherheitskorridor wird dann unter Verwendung der Operation *Besetzen* belegt, der *Wechsel* zwischen den Flugzonen vorgenommen und schließlich der Sicherheitskorridor wieder freigegeben.

Abschließend beschreibt die Operation *Abarbeitung* das Entnehmen eines Flugzeuges aus der Anfragemenge und den abgesicherten Wechsel dieses Flugzeuges über einen Sicherheitskorridor.

Flugraum

Flugraum

$flugzonen : Koordinate \rightarrowtail\!\!\!\!\!\!\; \leftrightarrow Flugzone$
$sicherheitskorridore : \mathbb{F}\ Sicherheitskorridor$
$anfragen : \mathbb{F}\ Flugzeug$

INIT

AnfrageAufnehmen

$\Delta(anfragen)$
$f? : Flugzeug$

AnfrageEntnehmen

$\Delta(anfragen)$
$f! : Flugzeug$

$Wechseln \mathrel{\widehat{=}}$

$AbgesichertWechseln \mathrel{\widehat{=}}$

$Abarbeitung \mathrel{\widehat{=}}$

Aufgabe 3 (Statecharts) - Automatisches Fensterhebesystem mit Kindersicherung im Auto

(Geschätzte Bearbeitungszeit: 20 min.)

In modernen viertürigen Autos wird das Öffnen und Schließen der 4 Fenster durch ein elektronisches System geregelt.

An jeder der 4 Türen befindet sich ein Fensterschalter mit drei möglichen Stellungen:

- In der *Ruhelage* wird die Position des zugehörigen Fensters nicht verändert,
- Durch Bewegen in die Stellung *Öffnen* wird das Fenster so lange kontinuierlich geöffnet, bis der Schalter losgelassen wird und in die Ruhelage zurückkehrt,
- Durch Bewegen in die Stellung *Schließen* wird das Fenster so lange kontinuierlich geschlossen, bis der Schalter losgelassen wird und in die Ruhelage zurückkehrt.

An der Fahrertür befinden sich zusätzlich zwei weitere Schalter der gleichen Bauart:

- Zentralschalter, mit ihm können alle 4 Fenster gleichzeitig geöffnet und geschlossen werden,
- Beifahrerschalter, mit ihm kann nur das Fenster an der Beifahrertür vom Fahrer geöffnet und geschlossen werden.

Außerdem gibt es eine zentrale Vorrichtung, die es ermöglicht, eine Kindersicherung ein- oder auszuschalten. Bei eingeschalteter Kindersicherung wird das Öffnen der hinteren beiden Fenster verhindert, wenn einer der Fensterschalter betätigt wird. Nur bei Betätigung des Zentralschalters ist die eingeschaltete Kindersicherung nicht wirksam.

Stellen Sie sicher, dass beim Einschalten der Zündung alle 4 Fenster geschlossen sind.

Aufgabenstellung

20 Punkte

Modellieren Sie das beschriebene System mit Hilfe von Statecharts. Beschreiben Sie zusätzlich anhand der vorgegebenen Tabelle alle auftretenden Ereignisse kurz.

Verwenden Sie diese Tabelle zur Beschreibung der Ereignisse ihres Statecharts.

NAME	BESCHREIBUNG
AUSGABEEREIGNISSE	
EINGABEEREIGNISSE	
INTERNE EREIGNISSE	

*Klausur SWT 23.7.2003, Name:*__________________________ *MatrNr.:* __________

Bitte Verwendung dieses Blattes ankreuzen:

☐ Bearbeitung von Aufgabe _____

☐ Schmierzettel

Bitte Verwendung dieses Blattes ankreuzen:

☐ Bearbeitung von Aufgabe _____

☐ Schmierzettel

Bitte Verwendung dieses Blattes ankreuzen:

☐ Bearbeitung von Aufgabe _____

☐ Schmierzettel

*Klausur SWT 23.7.2003, Name:*__________________________ *MatrNr.:* __________

Bitte Verwendung dieses Blattes ankreuzen:

☐ Bearbeitung von Aufgabe _____

☐ Schmierzettel

*Klausur SWT 23.7.2003, Name:*____________________________ *MatrNr.:* ___________

Bitte Verwendung dieses Blattes ankreuzen:

☐ Bearbeitung von Aufgabe _____

☐ Schmierzettel