Technische Universität Berlin
Institut für Kommunikations- und Softwaretechnik
FG Softwaretechnik
Franklinstr. 28/29
10587 Berlin

Jähnichen
Anlauff
Bittner
Weber

Klausur

Softwaretechnik und Systemgestaltung

19. Juli 1996

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 **20 Punkte** notwendig. Insgesamt sind 50 Punkte möglich. Die Bearbeitungszeit beträgt 120 Minuten. In der Klausur sind **sämtliche Hilfsmittel** zugelassen.

Viel Erfolg!

Name, Vorname: ______________________________

Matrikelnummer: ________________

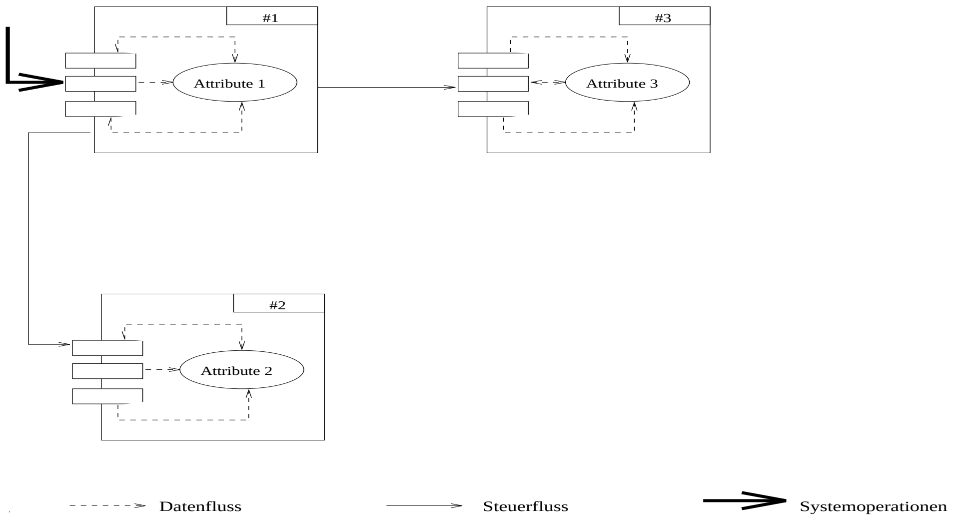
Studienrichtung/Prüfungsordnung: ________________
SSG-Übungen habe ich im Semester ____________besucht.

Aufgabe	1	2	3	4	5	Gesamt
Maximal	10	12	8	14	6	50
Erreicht:						

Aufgabe 1 (Allgemeines)

(Geschätzte Bearbeitungszeit: 20 min.)

1. Skizzieren Sie graphisch das objektorientierte Ausführungsmodell anhand von 3 Objekten und geben Sie die Eigenschaften dieses Ausführungsmodells in Stichworten an. *3 Punkte*



- hohe Laufzeitkomplexität
- große Anzahl von Objekten
- dynamisches Erzeugen und Löschen von Objekten
- dynamische Kommunikationspfade

2. Zählen Sie die Phasen des Fusion-Prozesses auf und beschreiben Sie die zu jeder Phase gehörenden Modelle. *3 Punkte*

- Analyse
  - Objektmodell
  - Time-line Diagramme
  - Systemobjektmodell
  - Ablaufmodell
  - Operationsmodell

    - Datenlexikon

- Entwurf
    - Objektinteraktionsgraphen
    - Sichtbarkeitsgraphen
    - Klassenbeschreibungen
    - Vererbungsgraphen
    - Datenlexikon
- Implementierung
    - Code

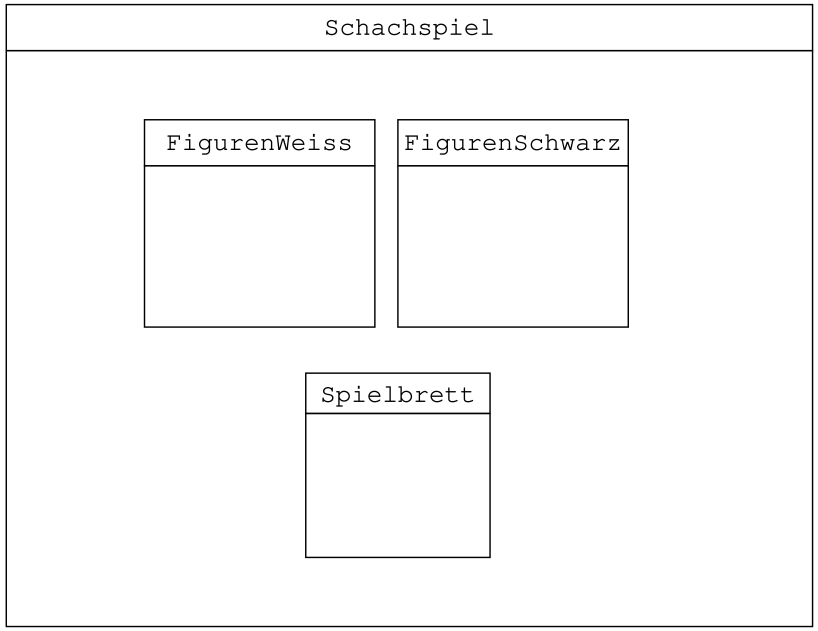
3. Beschreiben Sie Struktur und Aufbau eines Schachspiels gemäß nachfolgender Beschreibung. Verwenden Sie dazu die Klassen-, Vererbungs- und Aggregationsnotationen nach der Fusion Methode (keine Relationen!). *4 Punkte*

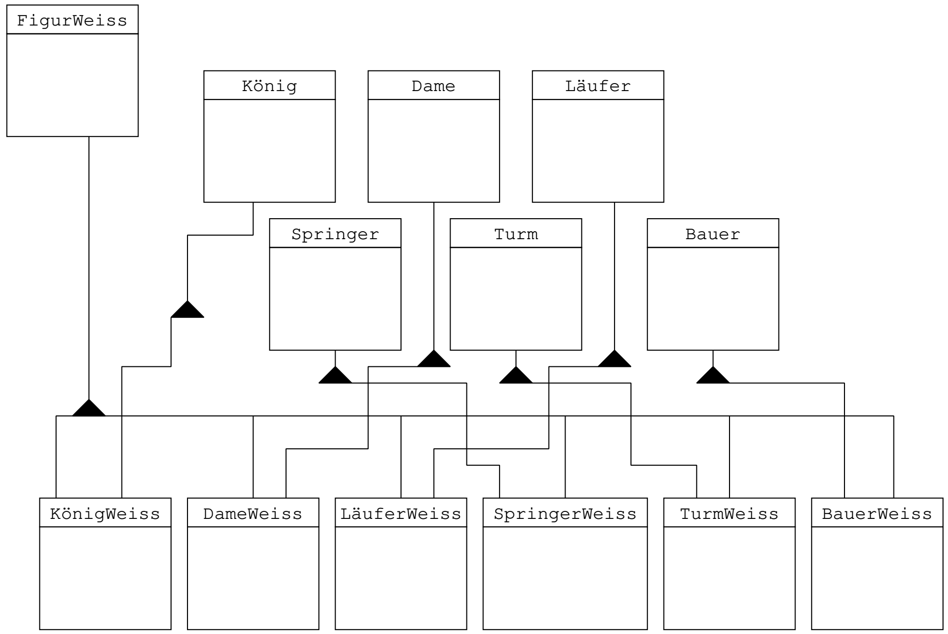
    Ein Schachspiel besteht aus Schachfiguren und aus einem Spielbrett. Es gibt einen Figurensatz mit ausschließlich weißen Figuren und einen Figurensatz mit ausschließlich schwarzen Figuren. Man unterscheidet zwischen den Figuren König, Dame, Läufer, Springer, Turm, Bauer. Ein Figurensatz besteht jeweils aus:

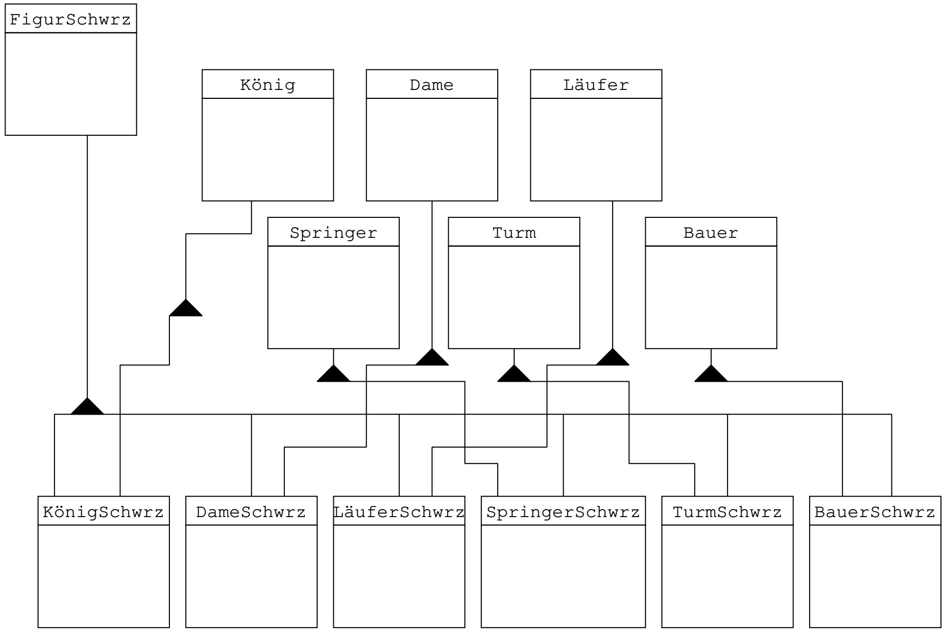
    - *1 König*
    - *1 Dame*
    - *2 Läufer*
    - *2 Springer*
    - *2 Türme*
    - *8 Bauern*

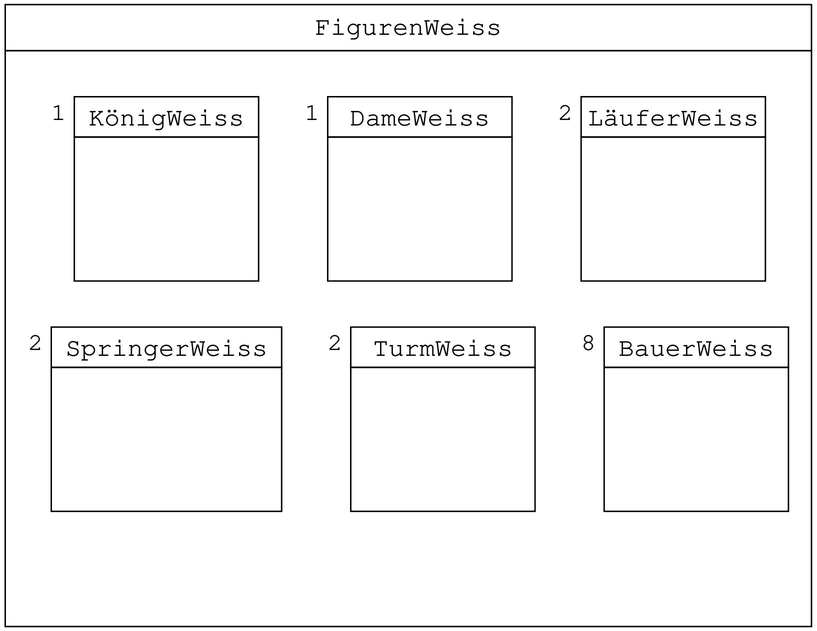
    Unterteilen Sie Ihr Modell in mehrere Teilmodelle!

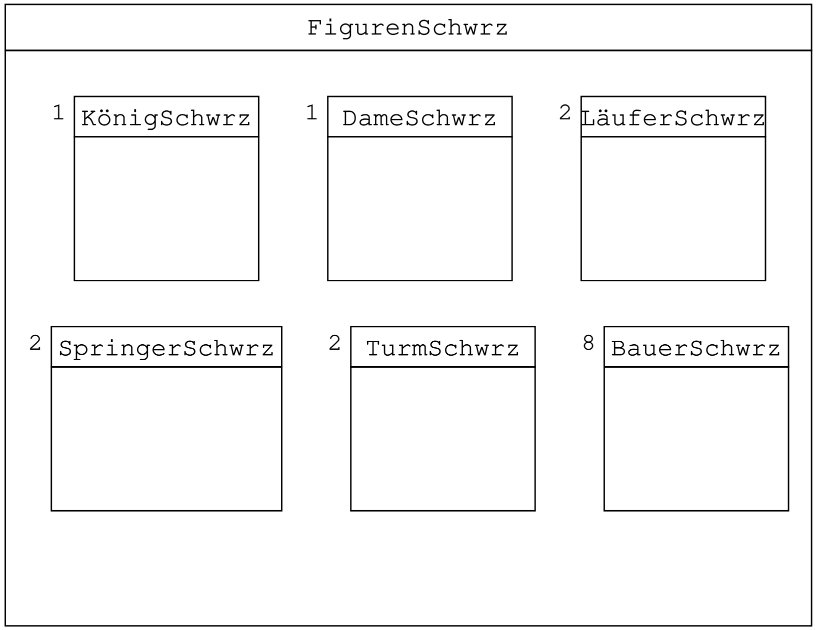
Lösung:











Aufgabe 2 (Fusion)

(Geschätzte Bearbeitungszeit: 25 min.)

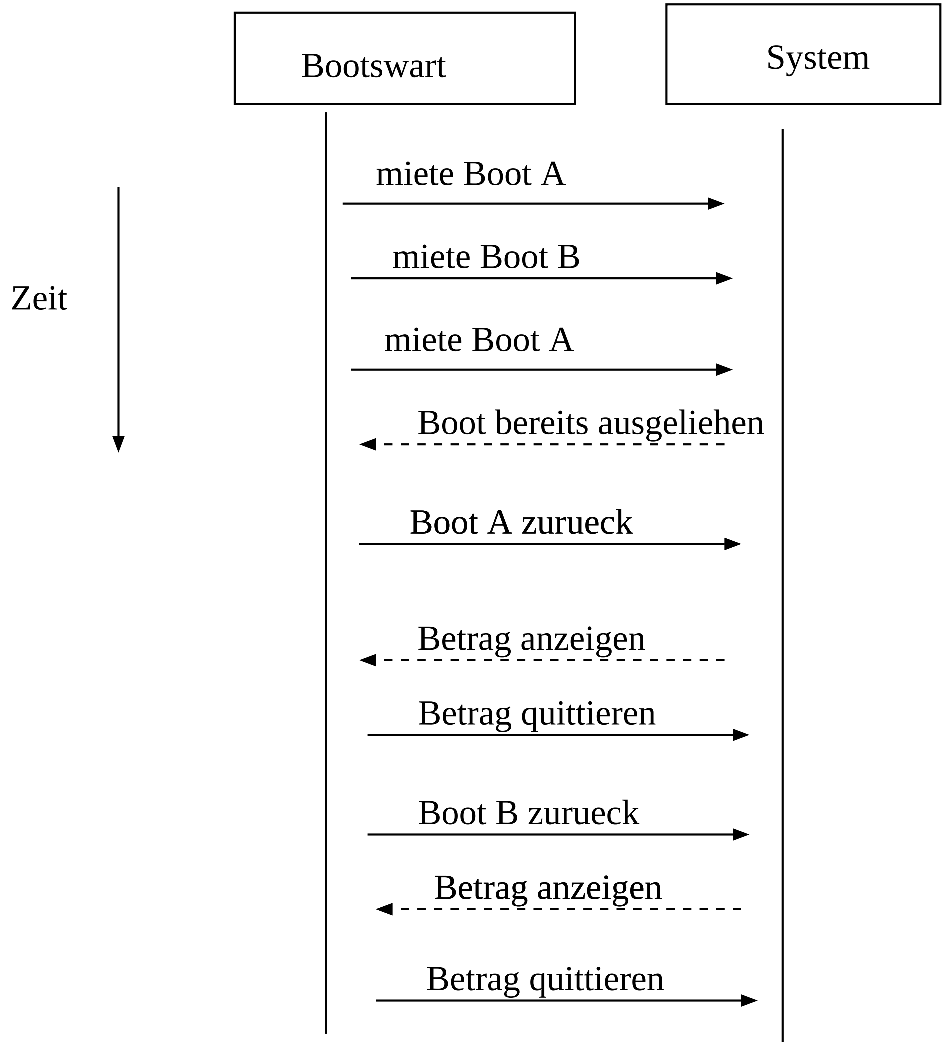
Am einem See gibt es einen Bootsverleih, der Boote für eine bestimmte Zeit vermietet. Die Ausleihe der Boote soll von einem EDV-System verwaltet werden. Es stehen verschiedene Bootstypen zur Auswahl: Tret-, Ruder-, Motor- oder Segelboot). Der Kunde geht zum Bootswart und erkundigt sich, ob ein Boot frei ist. Wenn ja, macht der Bootswart das Boot klar, hilft dem Kunden ins Boot und gibt im System das Boot ein, das gemietet wurde. Im System wird der Vorgang „Bootsausleihe“ gestartet und darin die Startzeit vermerkt. Wenn der Kunde das Boot zurückbringt, gibt der Bootswart die Beendigung der Bootsausleihe ein. Das System zeigt daraufhin den Betrag an, die der Kunde für die Bootsausleihe bezahlen muß. Gibt der Bootswart ein bereits ausgeliehenes Boot im System ein, so schickt das System die Meldung `Boot_ist_bereits_ausgeliehen` zurück.

Das System gibt außerdem Auskunft über zur Zeit nicht ausgeliehene Boote.

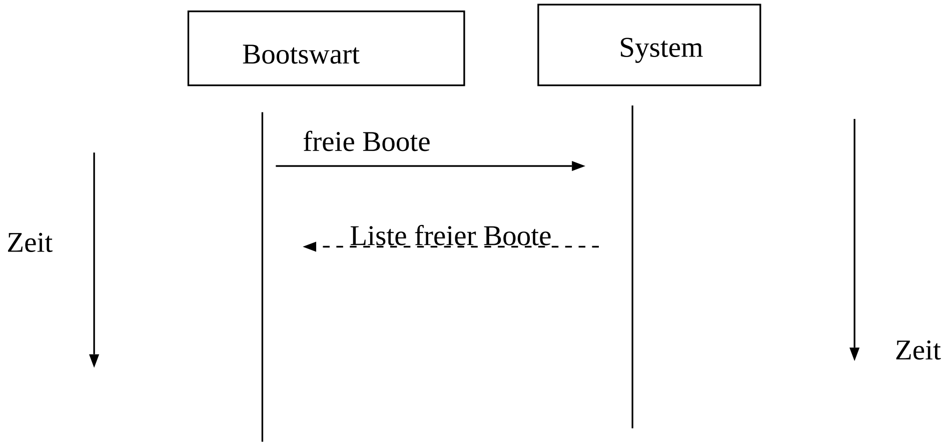
1.) Entwerfen Sie Szenarien (Time-Line Diagramme) für die folgenden Fälle:

a) Vorgang Bootsausleihe. Betrachten Sie die Bootsausleihe für zwei *3 Punkte*

Boote (alle möglichen Fälle). Verwenden Sie exemplarisch das Boot A und Boot B.



b) Der Bootswart stellt an das System die Anfrage, welche Boote zur Zeit nicht ausgeliehen sind. *3 Punkte*



2.) Vervollständigen Sie anhand des vorgegebenen Operationsschemas einen Objektinteraktionsgraphen für die Systemoperation **Boot_mieten**. *6 Punkte*

Operationsschema „`Boot_mieten`“

Operation: Boot_mieten

Description: Bootswart informiert das System, daß ein Boot gemietet wurde
Boote werden über ihren Namen identifiziert

Reads: **supplied** name // Name des Boots
boot **with** boot.name = name, uhr.zeit

Changes: **new** bootsausleihe, terminal

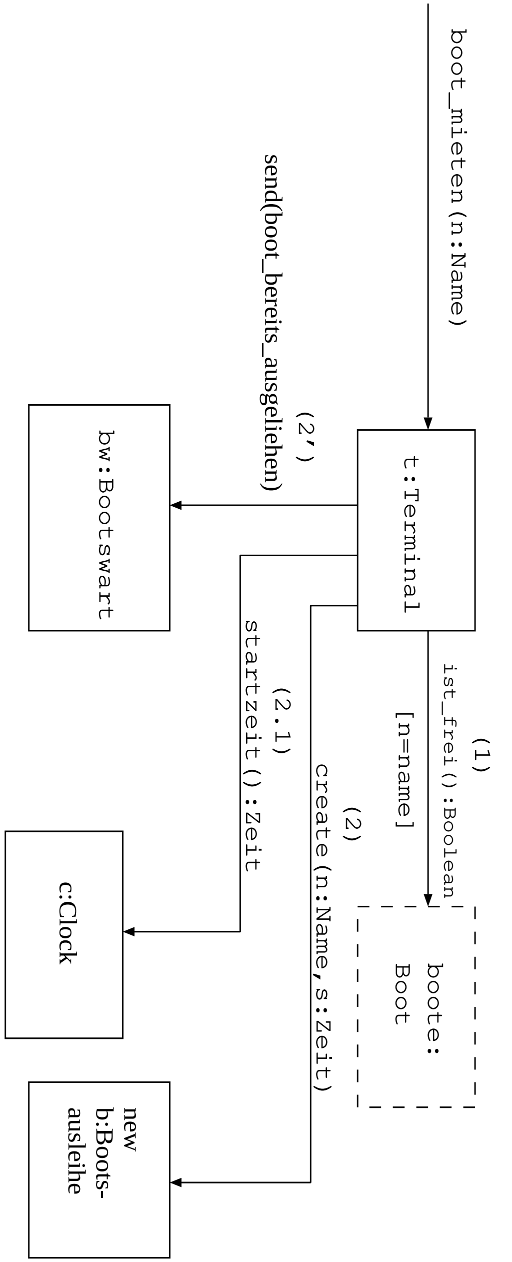
Sends: Bootswart:{Boot bereits ausgeliehen}

Assumes:

Result: **if** (Boot bereits ausgeliehen)
then
Bootswart wurde informiert Boot bereits ausgeliehen
otherwise
Eine neue Bootsausliehe wurde erzeugt.
Bootsausleihe.start wurde auf uhr.time gesetzt.

Gegeben ist der Controller *Terminal*. Sequentalisieren Sie die Methodenaufrufe.

Lösung:

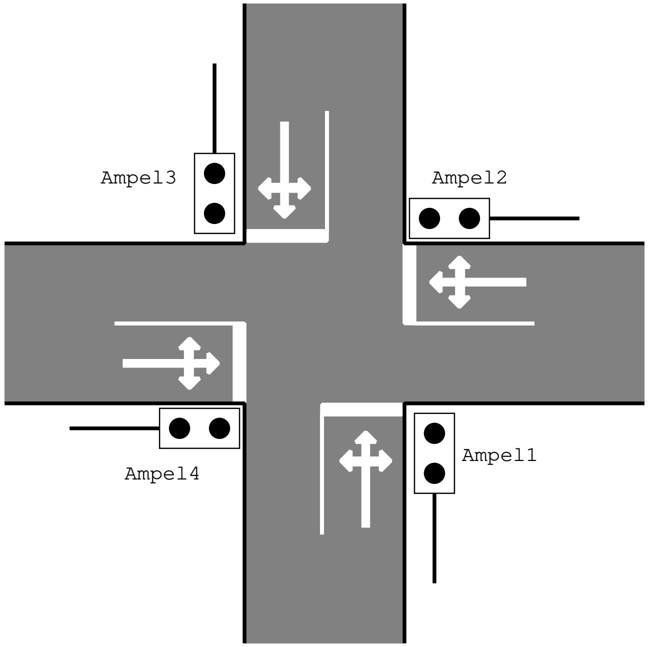


Aufgabe 3 (Petri-Netze)

(Geschätzte Bearbeitungszeit: 15 min.)

8 Punkte

Gegeben sei eine Verkehrskreuzung (siehe Abbildung), die durch Ampeln geregelt wird. Die Ampeln können entweder „Grün“ (freie Fahrt) oder „Rot“ (anhalten) anzeigen. Die Ampeln sind so geschaltet, daß die Ampeln 1 und 3 jeweils das gleiche Signal anzeigen. Dies gilt auch für die Ampeln 2 und 4.



Beachten Sie, daß nicht alle Ampeln gleichzeitig auf „Grün“ geschaltet sind. Dabei haben die Stellen und Transitionen folgende Bedeutung:

- s_1: Ampeln 1 und 3 zeigen „Rot“ an.
- s_2: Ampeln 1 und 3 zeigen „Grün“ an.
- s_3: Ampeln 2 und 4 zeigen „Rot“ an.
- s_4: Ampeln 2 und 4 zeigen „Grün“ an.

t_1: Ampeln 1 und 3 schalten von „Rot“ auf „Grün“.

t_2: Ampeln 1 und 3 schalten von „Grün“ auf „Rot“.

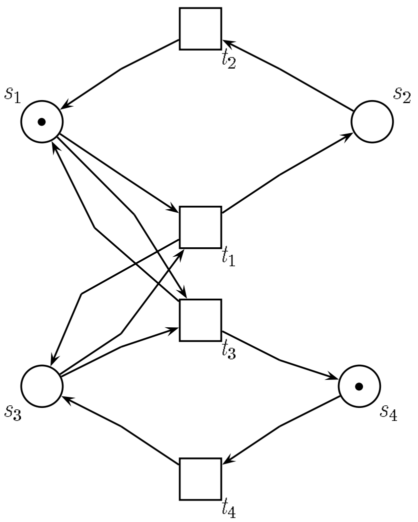
t_3: Ampeln 2 und 4 schalten von „Rot“ auf „Grün“.

t_4: Ampeln 2 und 4 schalten von „Grün“ auf „Rot“.

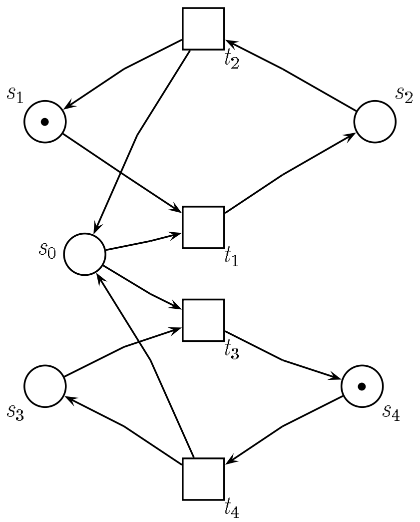
Vervollständigen Sie entsprechend das vorgegebene Petri-Netz.

Hinweis: Beachten Sie, daß Sie **nicht die Steuerung** der Ampeln modellieren sollen, sondern lediglich eine Überwachung der Ampelzustandsübergänge.

Lösung 1:



Lösung 2:



Aufgabe 4 (Z)

(Geschätzte Bearbeitungszeit: 30 min.)

Das Buchungssystem eines Konzerthauses registriert die Buchungen der verfügbaren Plätze. Der Konzertsaal hat 15 Reihen mit je 30 Sitzen. Ein Kunde kann einen oder auch mehrere Plätze auf seinen Namen buchen lassen. Der Zustandsraum sei mit dem folgendem Z Schema $Buchungssystem$ beschrieben (die Notation $1 \mathinner{..} n$ steht für $\{k : \mathbb{N} \mid 1 \leq k \leq n\}$).

$[PERSON]$

$Buchungssystem$

$Kunden : \mathbb{F}\, PERSON$

$gebucht : (\mathbb{N} \times \mathbb{N}) \pinj PERSON$

$\mathrm{dom}\; gebucht \subseteq (1 \mathinner{..} 15) \times (1 \mathinner{..} 30)$

$\mathrm{ran}\; gebucht = Kunden$

a) Zur Kassenöffnung liegen keine Buchungen vor. Beschreiben Sie die Initalisierungsoperation $Init$ für das Buchungssystem. *2 Punkte*

Lösung:

$Init$

$Buchungssystem'$

$gebucht' = \varnothing$

b) Beschreiben Sie die Operation $Buchen$, die für einen Kunden einen (beliebigen) freien Platz, spezifiziert durch Reihennummer und Sitznummer, ausgibt und gleichzeitig bucht. Betrachten Sie nur den Normalfall! *3 Punkte*

Lösung:

Buchen

$\Delta Buchungssystem$
$k? : PERSON$
$r!, s! : \mathbb{N}$

$(r!, s!) \notin \operatorname{dom} gebucht$
$1 \leq r! \leq 15$
$1 \leq s! \leq 30$
$gebucht' = gebucht \cup \{(r!, s!) \mapsto k?\}$
$Kunden' = Kunden \cup \{k?\}$

c) Beschreiben Sie die Operation $Stornieren$ zum Löschen aller für einen Kunden existierenden Buchungen. Betrachten Sie nur den Normalfall! *3 Punkte*

Lösung:

Stornieren

$\Delta Buchungssystem$
$k? : PERSON$

$k? \in Kunden$
$gebucht' = gebucht \rhd \{k?\}$
$Kunden' = Kunden \setminus \{k?\}$

d) Beschreiben Sie die Operation $Anfrage$, die eine gewünschte Anzahl von *zusammenhängenden* freien Plätzen innerhalb einer Reihe sucht. Die freien Plätze werden dann durch die Reihennummer sowie die Anfangs- und Endnummer der Sitze ausgegeben. Betrachten Sie nur den Normalfall! *6 Punkte*

Lösung:

Anfrage

$\Xi Buchungssystem$
$a? : \mathbb{N}$
$r!, s1!, s2! : \mathbb{N}$

$s2! - s1! + 1 = a?$
$\forall\, i : s1! \,..\, s2! \bullet (r!, i) \notin \operatorname{dom} gebucht$

Hinweis: „Normalfall“ bedeutet die erfolgreiche Durchführung der Operation wie beschrieben. Betrachten Sie keine Fehler- oder Ausnahmesituationen.

Aufgabe 5 (Statecharts)

(Geschätzte Bearbeitungszeit: 10 min.)

Gegeben sei der unten abgebildete Statechart.

a) Angenommen, nach Initalisierung des Systems treten nacheinander die Ereignisse a, e, g, g, g, g, b, c auf. In welchem Zustand bzw. welchem Zustandspaar befindet sich dann das System. *3 Punkte*

Lösung: (D,J)

b) Angenommen, nach Initalisierung des Systems treten nacheinander die Ereignisse $a, g, g, h, b, a, a, a, a$ auf. In welchem Zustand bzw. welchem Zustandspaar befindet sich dann das System. *3 Punkte*

Lösung: (C,G)