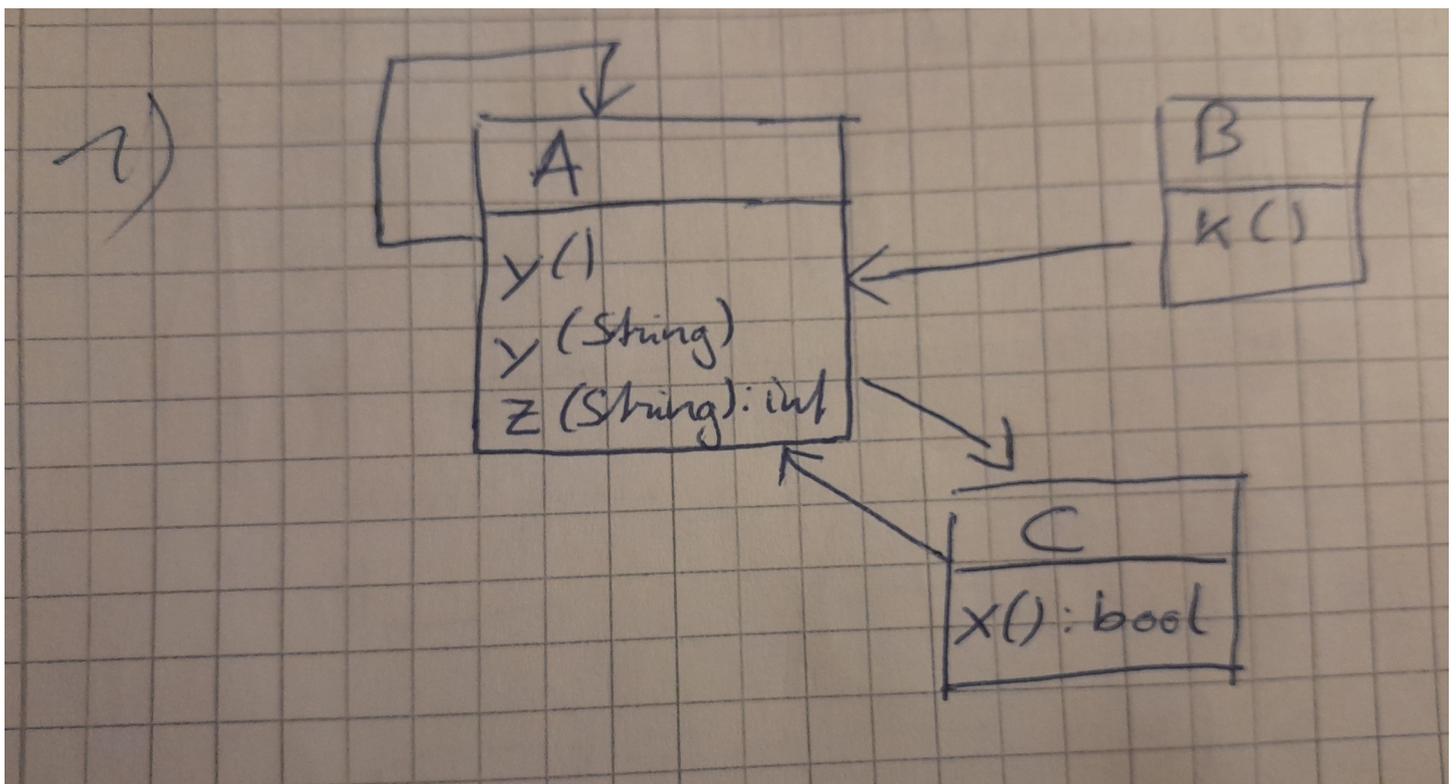
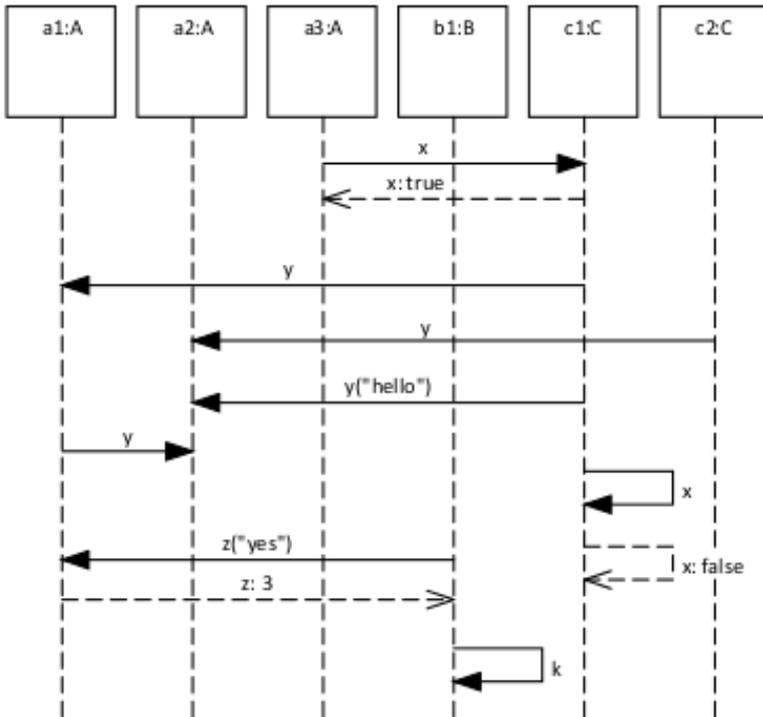


Aufgabe 1: Klassendiagramm aus Sequenzdiagramm

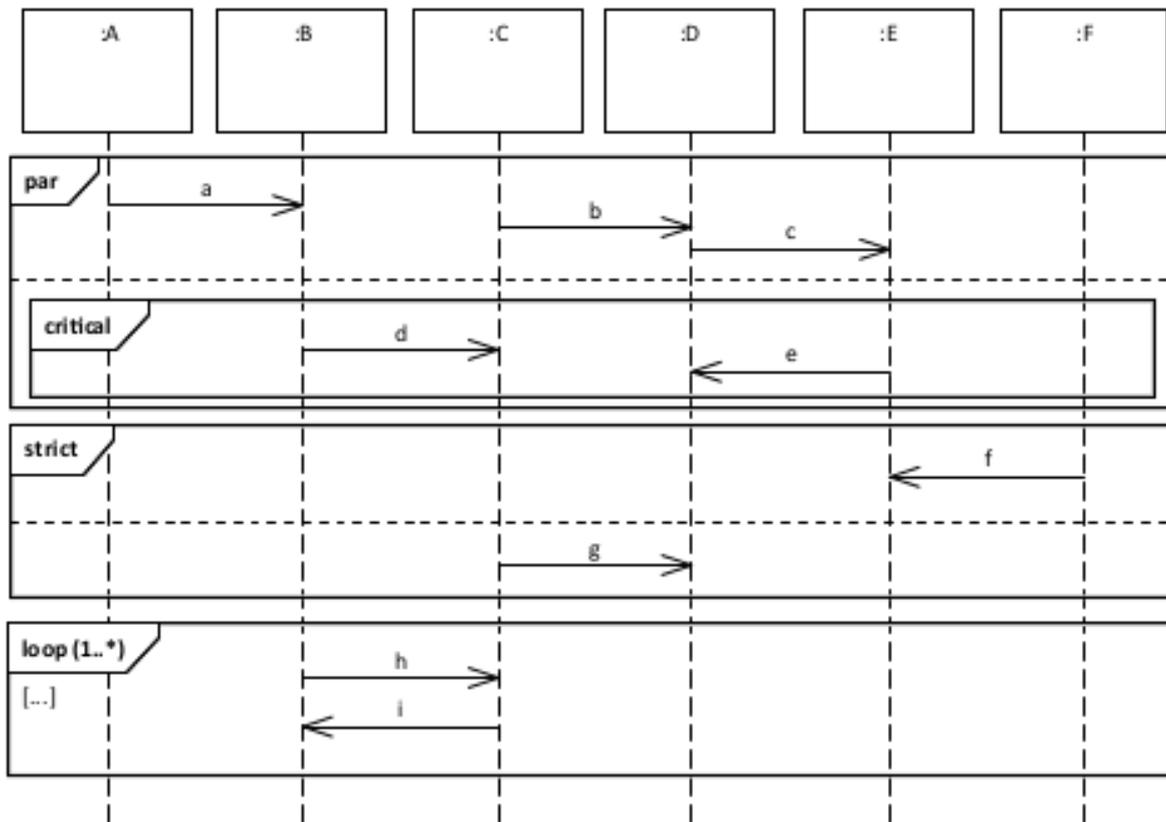
Gegeben ist das nachfolgende Sequenzdiagramm. Modellieren Sie das Klassendiagramm, auf dem das gegebene Sequenzdiagramm basiert.

- Klassen
- Operationsdefinitionen mit Typangaben, soweit ersichtlich
- Beziehungen zwischen Klassen in Form von navigierbaren Assoziationen: Zeichnen Sie nur Navigationsrichtungen ein, die aus dem gegebenen Sequenzdiagramm ersichtlich sind.



Aufgabe 2: Sequenzdiagramm: Berechnung von Traces

Gegeben ist das folgende Sequenzdiagramm:



- (a) Beschreiben Sie alle möglichen Ereignisfolgen des gegebenen Diagramms.
 (b) Welche der folgenden Traces sind möglich? Warum/warum nicht?

$b \rightarrow a \rightarrow c \rightarrow d \rightarrow e \rightarrow f \rightarrow g \rightarrow h \rightarrow i$	<input type="checkbox"/> richtig <input type="checkbox"/> falsch
$a \rightarrow b \rightarrow c \rightarrow d \rightarrow f \rightarrow e \rightarrow g \rightarrow h \rightarrow i$	<input type="checkbox"/> richtig <input type="checkbox"/> falsch
$a \rightarrow b \rightarrow d \rightarrow e \rightarrow c \rightarrow f \rightarrow g \rightarrow h \rightarrow i$	<input type="checkbox"/> richtig <input type="checkbox"/> falsch
$b \rightarrow c \rightarrow d \rightarrow a \rightarrow e \rightarrow f \rightarrow g \rightarrow h \rightarrow i$	<input type="checkbox"/> richtig <input type="checkbox"/> falsch
$a \rightarrow b \rightarrow c \rightarrow d \rightarrow e \rightarrow g \rightarrow f \rightarrow h \rightarrow i \rightarrow h \rightarrow i$	<input type="checkbox"/> richtig <input type="checkbox"/> falsch
$a \rightarrow b \rightarrow c \rightarrow d \rightarrow e \rightarrow f \rightarrow g$	<input type="checkbox"/> richtig <input type="checkbox"/> falsch
$a \rightarrow b \rightarrow e \rightarrow d \rightarrow c \rightarrow f \rightarrow g \rightarrow h \rightarrow i$	<input type="checkbox"/> richtig <input type="checkbox"/> falsch
$d \rightarrow e \rightarrow a \rightarrow b \rightarrow c \rightarrow g \rightarrow f \rightarrow h \rightarrow i$	<input type="checkbox"/> richtig <input type="checkbox"/> falsch
$c \rightarrow b \rightarrow a \rightarrow e \rightarrow d \rightarrow f \rightarrow g \rightarrow h \rightarrow i$	<input type="checkbox"/> richtig <input type="checkbox"/> falsch
$a \rightarrow b \rightarrow d \rightarrow e \rightarrow c \rightarrow f \rightarrow g \rightarrow i \rightarrow h$	<input type="checkbox"/> richtig <input type="checkbox"/> falsch

2) a) par: alle Varianten wo e und d direkt aufeinander
& critical folgen
 strict: f immer vorg
 loop 1..*: h -> i 1...n mal

b) f f + f f
 f + f f f

Aufgabe 3: Darstellung von Programmabläufen mittels Sequenzdiagramm

Stellen Sie die Abläufe von folgendem Programm mittels Sequenzdiagramm dar. Modellieren Sie auch allfällige Antwortnachrichten.

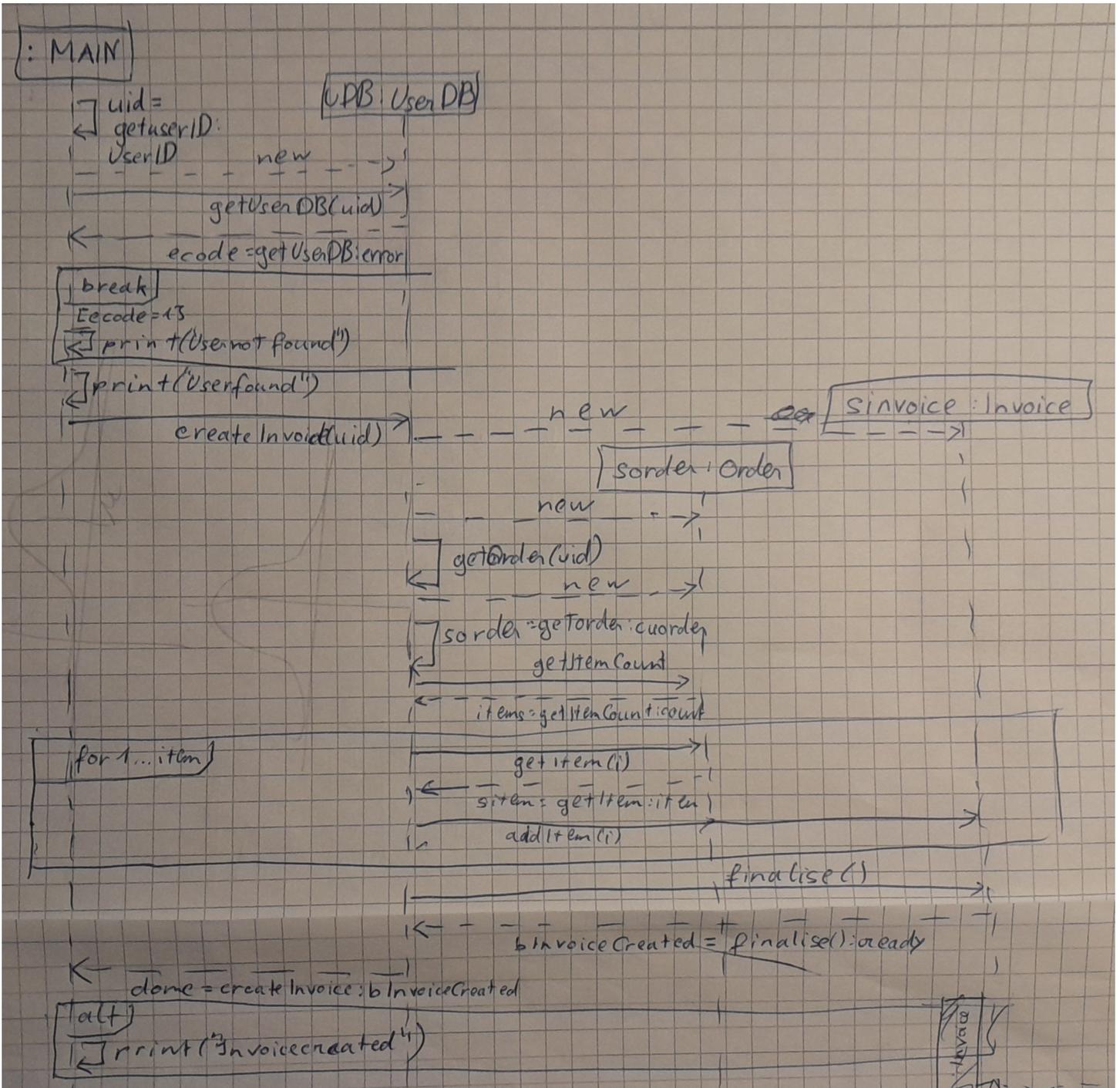
Sie können davon ausgehen, dass alle nicht explizit deklarierten Variablen bereits deklariert und initialisiert sind. „...“ markiert vernachlässigte Codeteile, die nicht modelliert werden müssen.

```
1 class Main {
2
3   public void main(String []) {
4     ...
5     String uid = getUserID();
6     UserDB uDB = new UserDB();
7     int ecode = uDB.getUserDB(uid);
8
9     if (ecode == 1) {
10      print("User not found!");
11      exit; //Programm wird beendet
12    }
13    else
14    {
15      print("User found");
16    }
17
18    boolean done = uDB.createInvoice(uid);
19
20    if (done) {
21      print("Invoice created!");
22    }
23    ...
24  }
25
26  public String getUserID() {
27    String UserID;
28    ...
29    return UserID;
30  }
31
32  public void print(String m) {...}
33 }
34
35 class UserDB {
36
37   public int getUserDB(String uid) {
38     int error;
39     ...
40     return error;
41   }
42
43   public Order getOrder(String uid) {
44     Order cuOrder = new Order();
45     ...
46     return cuOrder;
47   }
48
49   public boolean createInvoice(String uid) {
50     Invoice sinvoice = new Invoice();
```

```

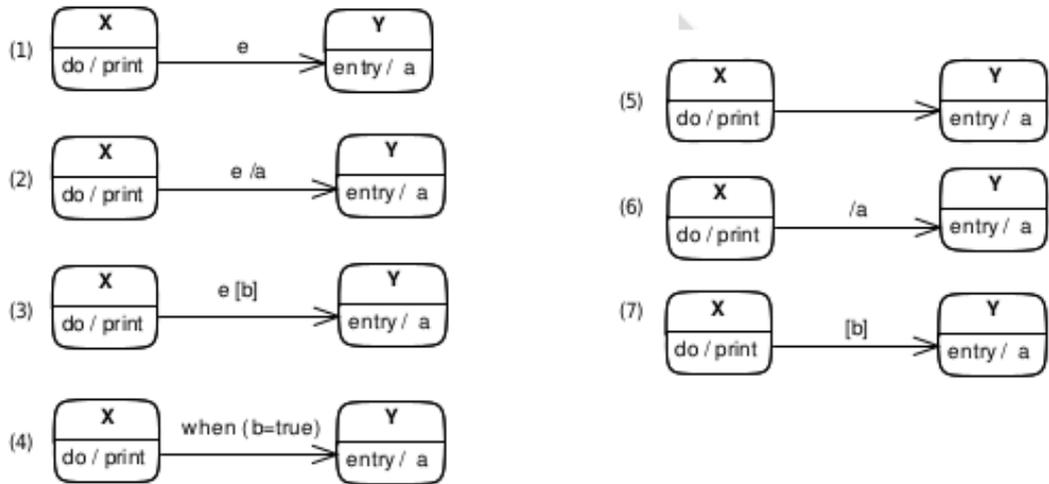
51     Order sOrder = new Order();
52     boolean blnInvoiceCreated = false;
53     String sItem = "";
54
55     sOrder = this.getOrder( uid );
56     int iItems = sOrder.getItemCount();
57
58     for (int i = 1; i <= iItems; i++) {
59         sItem = sOrder.getItem(i);
60         sInvoice.addItem(sItem);
61     }
62
63     blnInvoiceCreated = sInvoice.finalise();
64     ...
65     return blnInvoiceCreated;
66 }
67 }
68
69 class Order {
70
71     public int getItemCount() {
72         int count = 0;
73         ...
74         return count;
75     }
76
77     public String getItem(int num) {
78         String item = "";
79         ...
80         return item;
81     }
82 }
83
84 class Invoice {
85
86     public void addItem(String sItem) {
87         ...
88         return;
89     }
90
91     public boolean finalise() {
92         boolean ready = true;
93         ...
94         return ready;
95     }
96 }

```

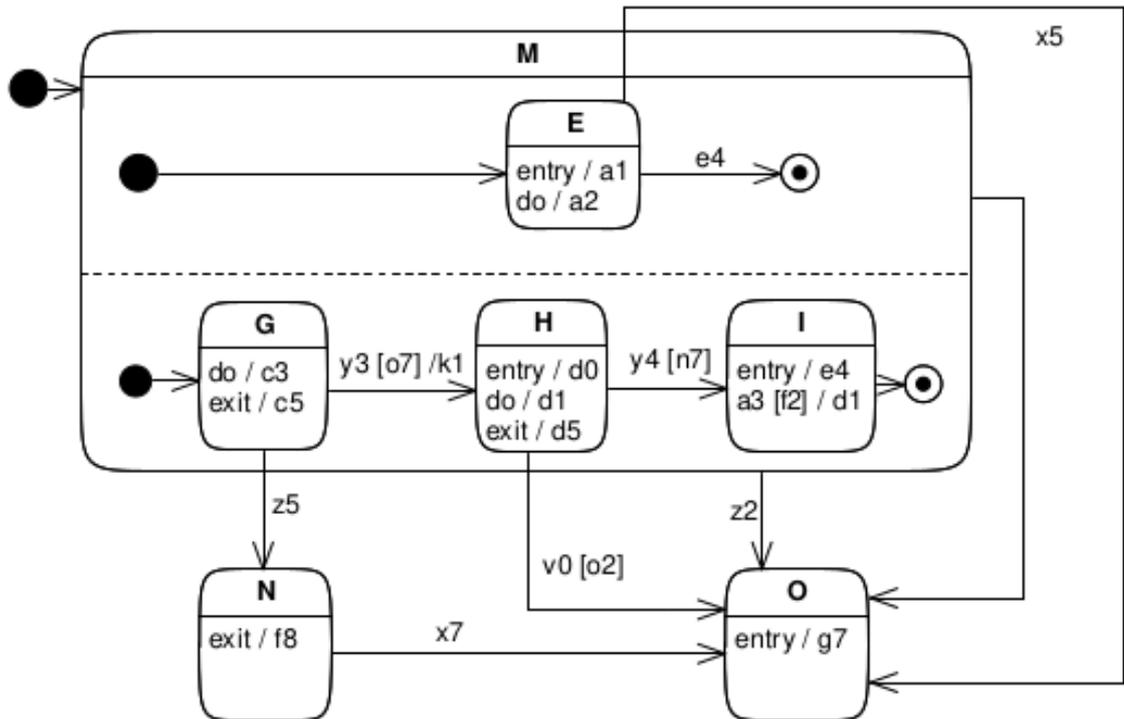


Aufgabe 4: Zustandsdiagramm: Zustände und Zustandsübergänge

a) Wann erfolgt eine Transition von einem Zustand in einen anderen? Nutzen Sie die nachfolgenden Ausschnitte, um die Frage zu beantworten.



b) Gegeben ist das nachfolgende Zustandsdiagramm.



- i) In welchen Kombinationen von Zuständen kann sich der Automat zu einem Zeitpunkt gleichzeitig befinden?
- ii) Wann erfolgt die Transition von Zustand G nach Zustand N? In welchem Zustand muss sich das System befinden, damit die Transition erfolgen kann? Welcher Zustand bzw. welche Zustände sind nach der Transition aktiv?
- iii) Wann bzw. unter welchen Voraussetzungen erfolgt die Transition von Zustand M nach Zustand O?

1) wenn e eintritt

5) wenn print fertig

2) wenn e eintritt

6) wenn print fertig

3) wenn e eintritt und b erfüllt

7) wenn print fertig und b erfüllt

4) wenn b wahr

b) i) $E \wedge (G \vee H \vee I)$

ii) wann $G \rightarrow N$? wenn z5 eintritt

in welchem Zustand System? $E \wedge G$

welche Zustände nach Transaktion abis? N (E abgebrochen)

iii) wenn System in Zustand E ist und x5 ^{e4} eintritt

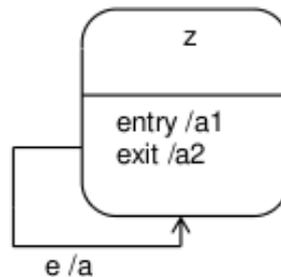
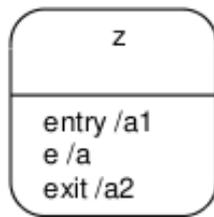
— i. i — list und ~~a3~~ ^{e4} eintritt und p? erfüllt

— ii — N ist und x7 eintritt

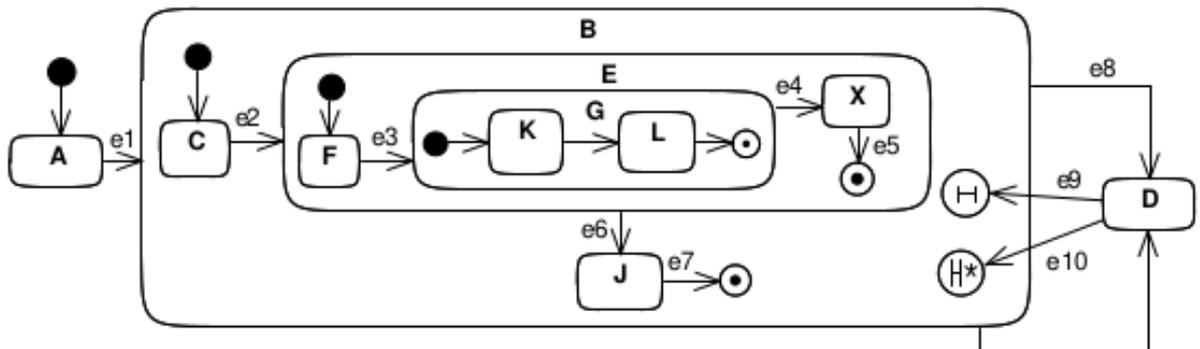
— iii — H ist und v0 eintritt und o2 erfüllt

— iv — M ist und z2 eintritt

c) Gegeben sind folgende zwei Ausschnitte eines Zustandsdiagramms. Sind die beiden Ausschnitte äquivalent? Begründen Sie Ihre Antwort!



d) Was versteht man unter einem historischen Zustand? Wann, warum und wie wird er eingesetzt? Benutzen Sie die nachfolgende Abbildung, um die Unterschiede zwischen flachem und tiefem historischem Zustand zu erklären.



c) Nein, links: entry/a1 \rightarrow e/a \rightarrow exit/a2
rechts: entry/a1 \rightarrow exit/a2 \rightarrow e/a \rightarrow entry/a1 \rightarrow exit/a2

d) Historische Zustände merken sich letzten internen Zustand
in einem komplexen Zustand bei Verlassen des Zustands
wenn in den Zustand zurückgekehrt wird geht das System
automatisch in den gemerkten Subzustand

flacher History Zustand: merkt sich 1 Ebene

tiefer — „ — : merkt sich mehrere Ebenen

anhand des Bsp: angenommen e1, e2, e3, e4, e8 merken

ein mit ^{den} dazugehörigen Zuständen, wenn dann e9 eintritt

würde ~~das~~ das System zu Zustand F zurückkehren,

wenn e 10 eintritt System zu Zustand X

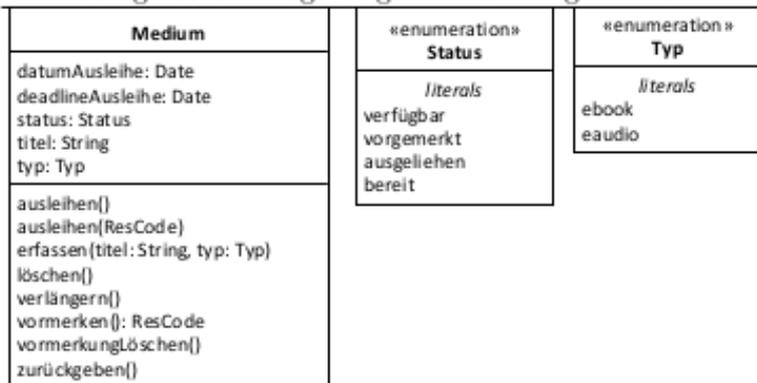
Aufgabe 5: Zustandsdiagramm: Virtuelle Bücherei

Ziel dieses Beispiels ist die Abbildung der Ausleihe von Medien in einer virtuellen Bücherei.

Sie haben folgende Informationen:

Zu Beginn muss jedes Medium erfasst werden. Medien können ausborgt werden, wobei ebooks für maximal 21 Tage und eaudios für maximal 7 Tage ausgeliehen werden können. Am Ende der Leihfrist werden sie automatisch zurückgegeben, sie können aber auch jederzeit früher zurückgegeben werden. Ausgeliehene Medien können vorgemerkt werden bzw. die Vormerkung kann auch wieder gelöscht werden. Die Leihfrist von einem nicht vorgemerkten ausborgten Medium kann einmalig um 7 Tage verlängert werden. Wird ein vorgemerktes Buch zurückgegeben bzw. läuft seine Leihfrist ab, so bleibt dieses für 48 Stunden „bereit“ zur Abholung. Wird es innerhalb dieser Frist nicht von der Person ausgeliehen, die es vorgemerkt hat, so ist es wieder für die allgemeine Ausleihe verfügbar. Ein Medium kann, wenn es nicht ausborgt ist, gelöscht werden. Dann sollen auch keine Informationen über das Medium mehr gespeichert sein.

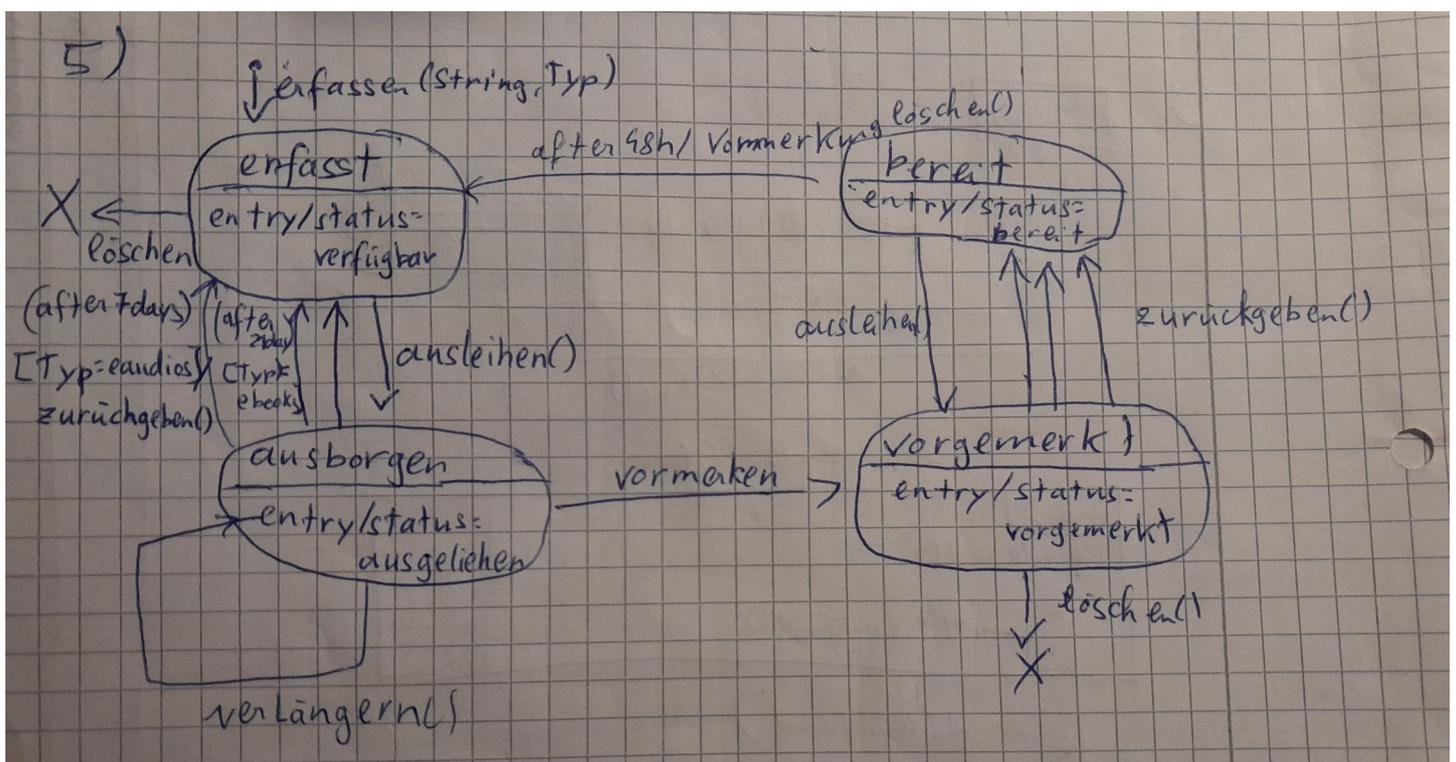
Ein Auszug aus dem zugehörigen Klassendiagramm sieht wie folgt aus:



Modellieren Sie ein Zustandsdiagramm, das den geschilderten Sachverhalt abbildet. Modellieren Sie dabei explizit alle Änderungen an den Attributen der Klasse „Medium“.

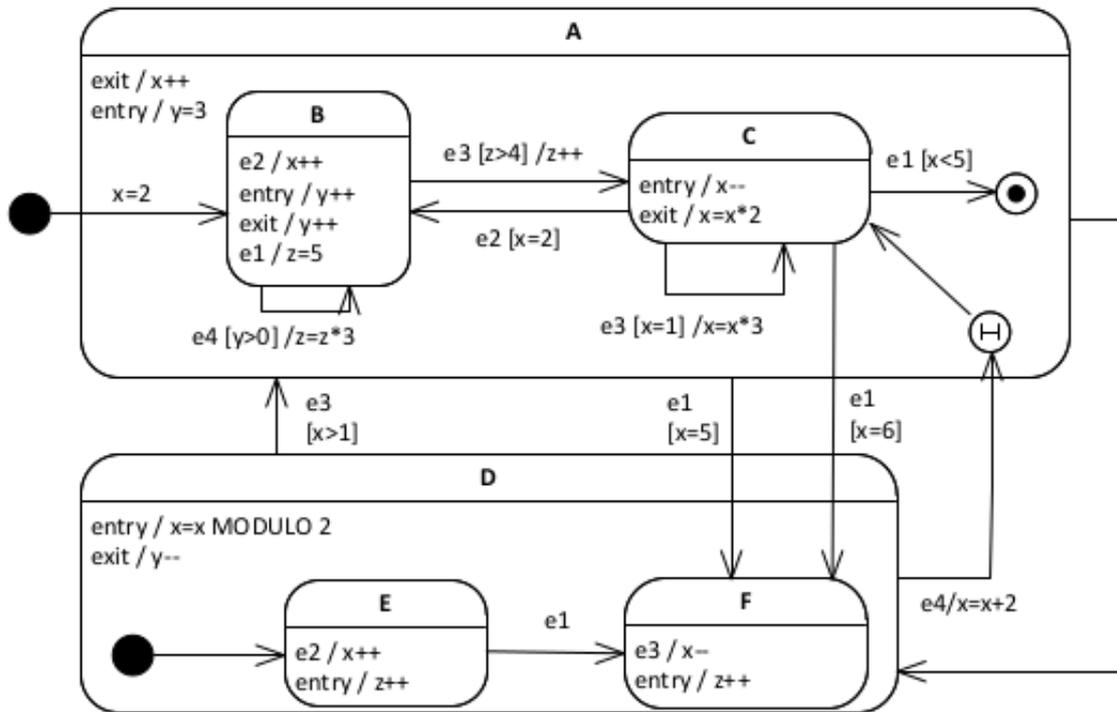
Betrachten Sie abschließend Ihr fertiges Zustandsdiagramm. Ist es übersichtlich und gut lesbar? Lässt es sich eventuell noch vereinfachen? Können Zustände zu komplexen (also zusammengesetzten) Zuständen zusammengefasst werden, um die Lesbarkeit zu erhöhen?

Überarbeiten Sie falls nötig Ihr Zustandsdiagramm, um es möglichst übersichtlich und lesbar zu gestalten.



Aufgabe 7: Zustandsdiagramm: Ereignisfolgen

Gegeben ist das nachfolgende Zustandsdiagramm. Vervollständigen Sie die zwei Tabellen, um zu veranschaulichen, welche Zustände und Aktionen bei den zwei Ereignisfolgen vorkommen.



70)

	eingeh. Zustand	x	y	z
Beginn	A / B	2	3/4	
e1	B			5
e3	C	1	5	6
e3	C	5		
e1	D / F	2/3/1		7
e4	A / C	3/2	4/3	

Beginn	Zust	x	y	z
Beginn	A / B	2	3/4	
e1	B			5
e3	C	1	5	6
e2				
e1	D / F	2/3/1		7
e2	E	2		

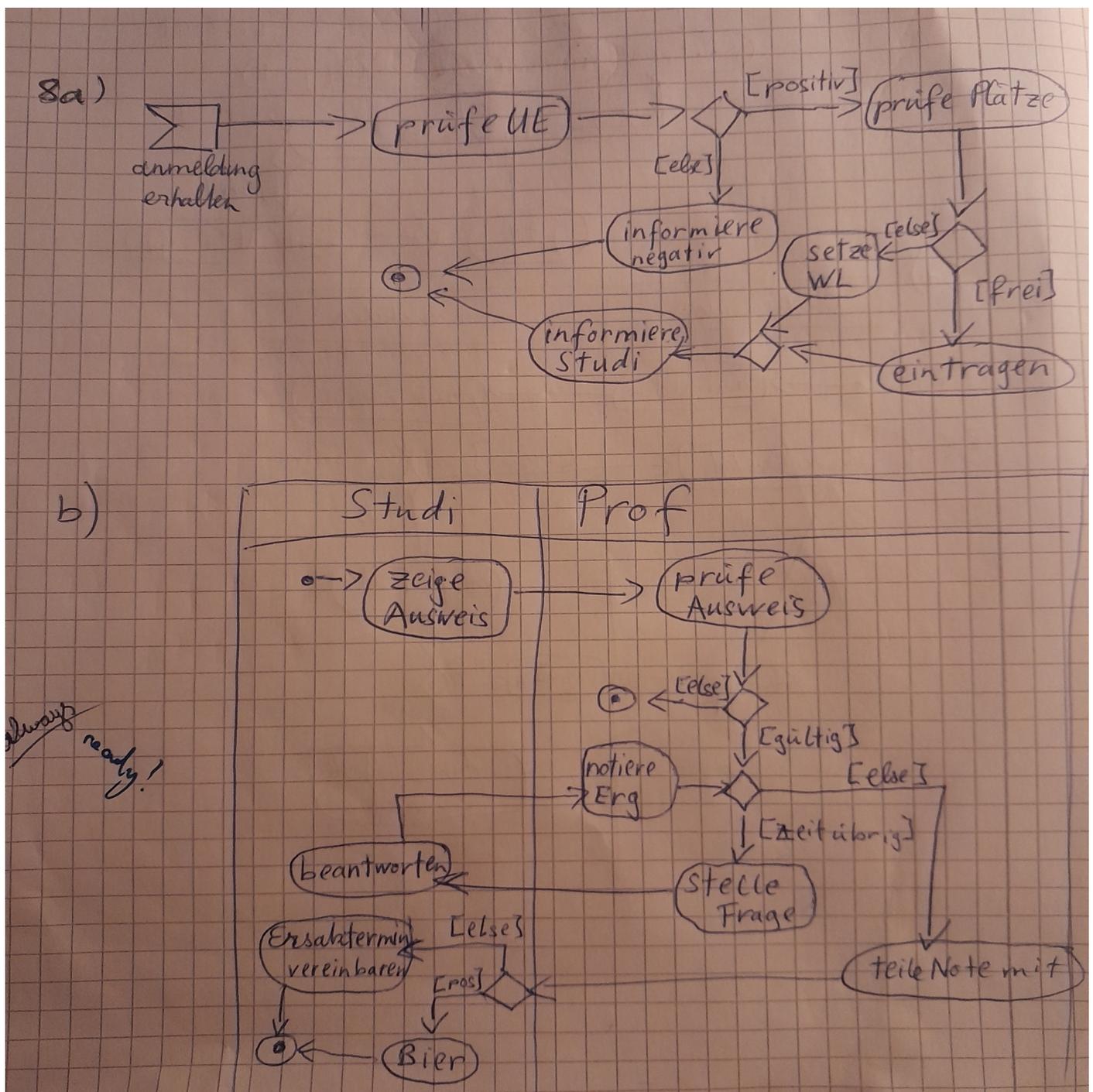
Aufgabe 8: Aktivitätsdiagramm: Einleitende Beispiele

a) Modellieren Sie folgenden Ablauf (Kontrollfluss) mittels Aktivitätsdiagramm:

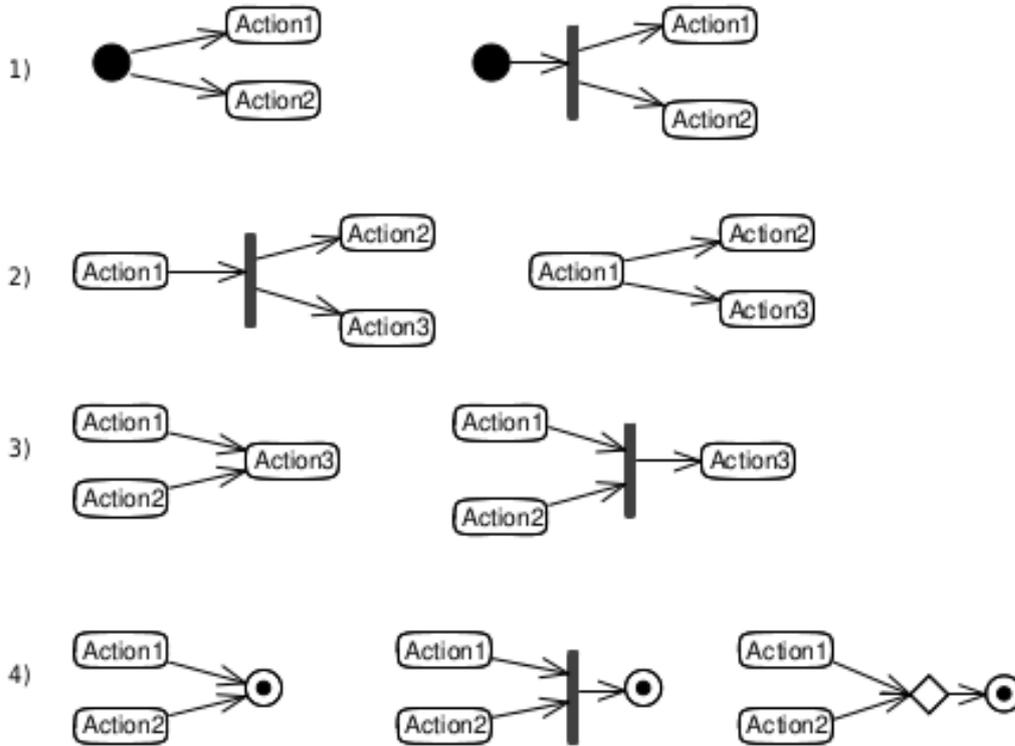
Ein/e Senior Lecturer erhält eine Anmeldung zu einer mündlichen Prüfung. Zunächst prüft er/sie, ob der/die Studierende eine positive Übungsleistung hat. Falls nicht, so informiert er/sie den/die Studierende darüber, dass ein Prüfungsantritt nicht möglich ist. Falls der/die Studierende beim Übungsteil positiv ist, so wird geprüft, ob noch Plätze für die Prüfung frei sind. Falls nein, wird der/die Studierende für den nächsten Termin vorgemerkt. Falls Plätze frei sind, so trägt er/sie den/die Studierende in die Anmeldeleiste ein. Abschließend wird der/die Studierende informiert.

b) Modellieren Sie folgenden Ablauf (Kontrollfluss) mittels Aktivitätsdiagramm:

Bei einer mündlichen Prüfung zeigt der/die Studierende dem/der Senior Lecturer zunächst den Ausweis, den er/sie daraufhin prüft. Ist der Ausweis nicht gültig, so ist der Prozess beendet. Andernfalls stellt der/die Senior Lecturer dem/der Studierenden eine Frage, die diese/r beantwortet und der/die Senior Lecturer protokolliert die Antwort. Anschließend wird die nächste Frage gestellt und wieder beantwortet und protokolliert. Dies wird solange wiederholt, bis keine Prüfungszeit mehr übrig ist. Nach der letzten Frage teilt der/die Senior Lecturer dem/der Studierenden die Note mit. Ist die Note negativ, so wird ein Ersatztermin vereinbart.



c) Sind folgende Konstrukte äquivalent bezüglich der Tokenverarbeitung?



d) Gegeben ist folgendes Aktivitätsdiagramm, das die Aktivitäten, die bei deiner Mundhygiene durchgeführt werden, zeigt:



Erweitern/Ändern Sie das Aktivitätsdiagramm so, dass folgende **Fehlersituationen** entsprechend behandelt werden:

- (i) Während der Reinigung mit Zahnseide reißt die Zahnseide. Daher muss ein Interdentalbürstchen geholt werden und stattdessen mit diesem gereinigt werden. Anschließend geht der Ablauf mit dem Entfernen des Zahnsteins regulär weiter.
- (ii) Die Schritte „Zahnstein entfernen“ bis „Polieren“ benötigen elektrischen Strom. Sollte es während dieser Schritte irgendwann zu einem Stromausfall kommen, so muss die Mundhygiene abgebrochen werden. Es wird ein neuer Termin vereinbart und der Prozess ist beendet.

Handwritten notes and diagram on grid paper:

1) ja, in beiden werden 2 neue Token generiert
 2) ja, an Action 2 und 3 werden jeweils ein Token weitergereicht
 3) ja, 2 Token werden zu 1
 4) nein, bei den Konstrukten links und rechts reißt 1 Token zu Beendigung des Vorgangs, bei dem id. M. 2 Token

d) i)

Aufgabe 9: Aktivitätsdiagramm aus Code

Gegeben sei der folgende Codeausschnitt. Modellieren Sie diesen als Aktivitätsdiagramm (nur den Kontrollfluss). Bilden Sie den Code möglichst genau ab.

Hinweis: Es handelt sich um einen Codeausschnitt. Variablendeklarationen, -initialisierungen etc. müssen nur modelliert werden, sofern diese auch im Codeausschnitt angeführt sind.

```
1 public static void main(String [] args) {  
2     int n = 10;  
3     for (int i = 1; i <= n; i++) {  
4         System.out.print(i + "\t");  
5         if (i%2==0){  
6             for (int j = 1; j <= n; j++) {  
7                 System.out.print("\t" + i * j);  
8             }  
9         }  
10        System.out.println ();  
11    }  
12 }
```

