Softwaretechnik
Klausur

(120 min/19.7.2011)

Nr: ___ **Name:** _____ **Matr.-Nr.:** _____

Aufgabe	mögliche Punkte	erreichte Punkte
1	6	
2	9	
3	10	
4	6	
5	6	
6	6	
7	12	
8	12	
9	12	
10	3	
11	10	
12	8	
Summe	100	

Benotungsskala	
Mindest-punktzahl	**Note**
94	1,0
88	1,3
82	1,7
76	2,0
70	2,3
64	2,7
58	3,0
52	3,3
46	3,7
40	4,0

Zum Bestehen der Klausur müssen mindestens 40 Punkte erreicht werden.

Note: _____

Aufgabe 1

(6 Punkte) Zeichnen Sie ein Anwendungsfalldiagramm, das das unten beschriebene Reisebürosystem (RBS) modelliert:

> Die Kunden eines Reisebüros können Reiseangebote ansehen oder Reiseangebote buchen. Das Buchen beinhaltet immer ein vorheriges Ansehen der Angebote. Premiumkunden können zusätzlich Reiseangebote reservieren. Auch eine Reservierung beinhaltet immer ein vorheriges Ansehen von Angeboten.
>
> Die Mitarbeiter eines Reisebüros können neue Angebote in das System einstellen und Reservierungen von Kunden in Buchungen umwandeln. Geschäftsführer können zusätzlich Angebote löschen.

Punkte: ____ /6

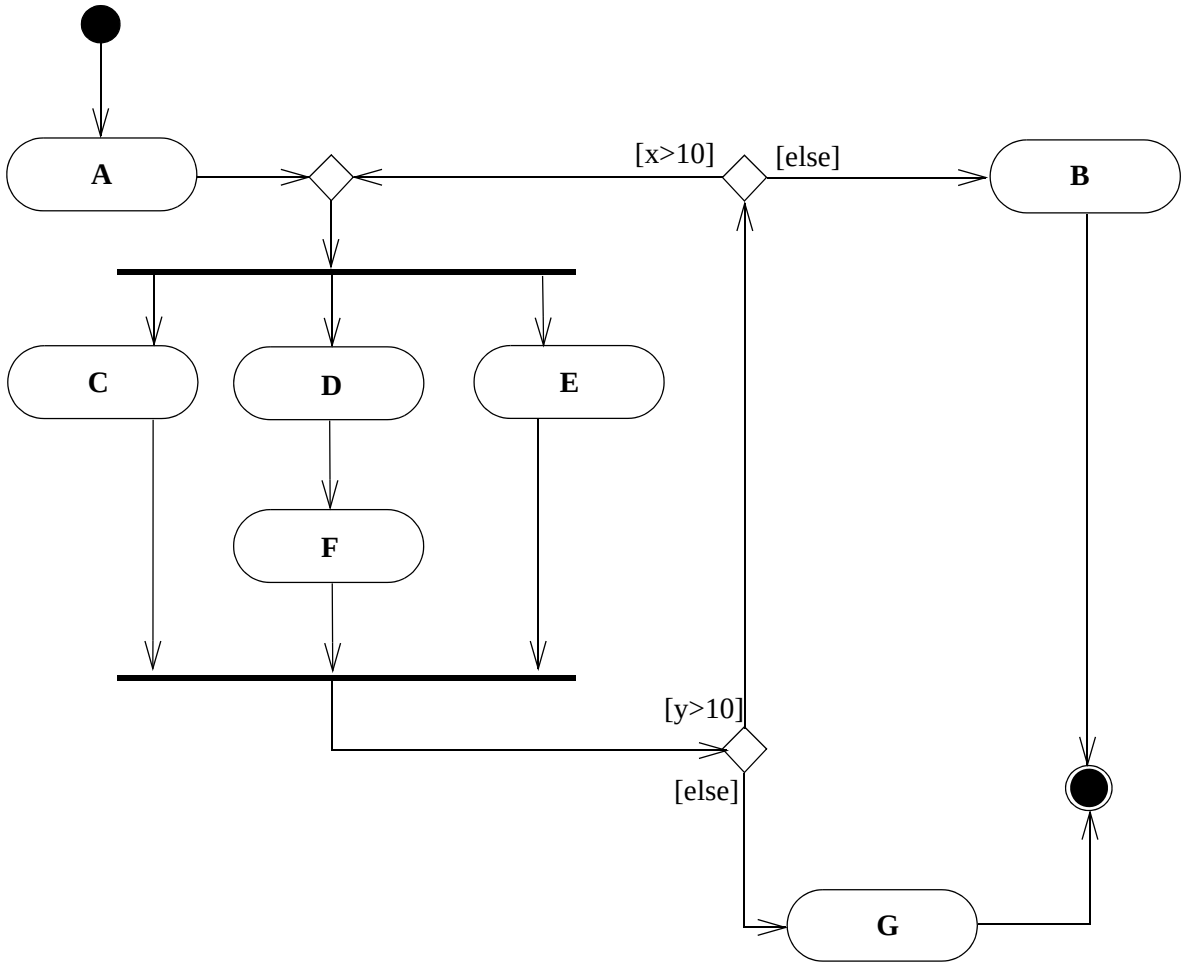
Aufgabe 2

Nehmen Sie in dem folgenden Aktivitätsdiagramm Veränderungen vor, so dass das beschriebene Verhalten ergänzt wird:

a) (2 Punkte) Wenn unmittelbar nach der Ausführung der Aktion **B** die Bedingung [a=0] gilt, soll mit der Aktion **G** fortgefahren werden. Gilt die Bedingung [a=0] nicht, soll sich der vorgegebene Ablauf nicht ändern.

b) (3 Punkte) Die Aktion **F** soll erst dann beginnen, wenn die Aktionen **D** und **E** beendet sind.

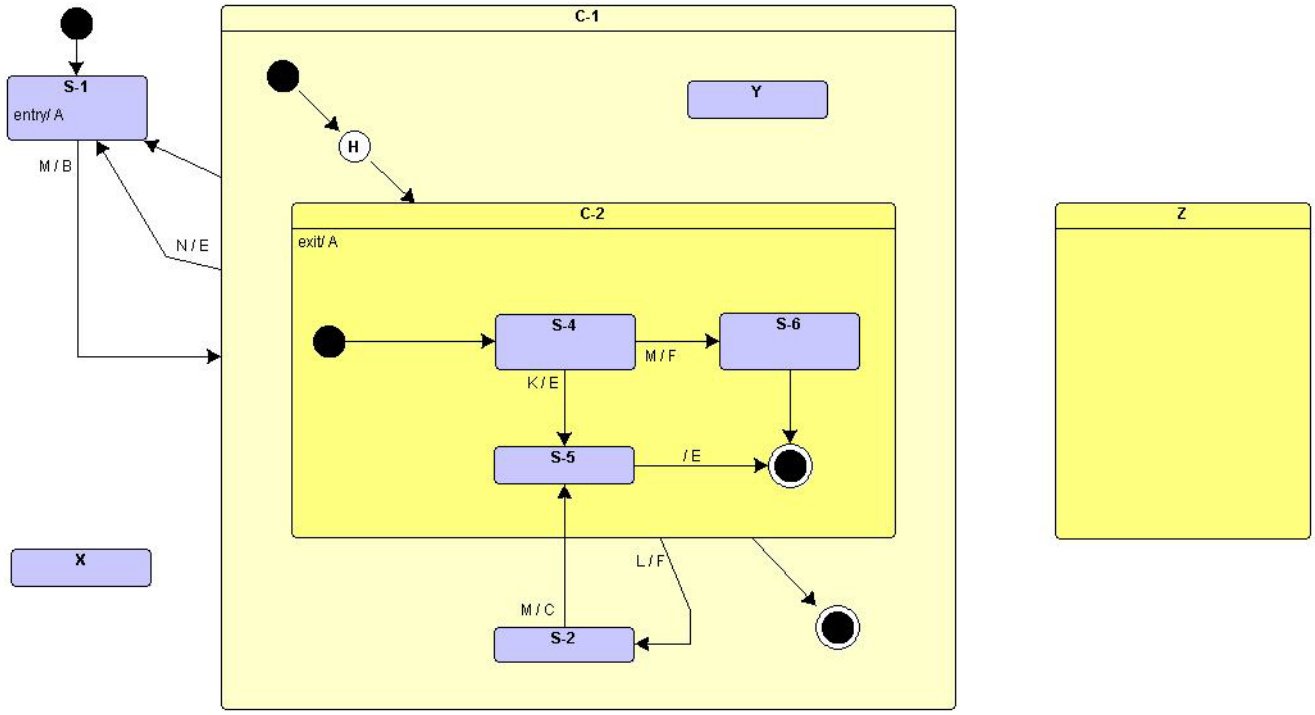
c) (4 Punkte) Wenn unmittelbar nach der Ausführung der Aktion **C** die Bedingung [b=0] gilt, sollen die Aktionen **D**, **E** und **F** unterbrochen werden und es soll direkt mit der Aktion **G** fortgefahren werden. Gilt die Bedingung [b=0] nicht, soll sich der vorgegebene Ablauf nicht ändern.



Punkte: ____ /9

Aufgabe 3

Gegeben sei das folgende Zustandsdiagramm:



a) (2 Punkte) Geben Sie an, welche Ausgabe durch die Eingabefolge **M L N M M** erzeugt wird.

Ausgabe: _____

Ergänzen Sie in dem oben stehenden Diagramm möglichst wenige Transitionen, um das beschriebene Verhalten zu erzeugen:

b) (2 Punkte) Aus jedem der Zustände aus **C-2** soll bei Eingabe von **P** in den Zustand **Y** gewechselt werden. Nur aus dem Zustand **Y** soll bei Eingabe von **P** wieder in genau den Zustand gewechselt werden, von dem aus **Y** erreicht wurde.

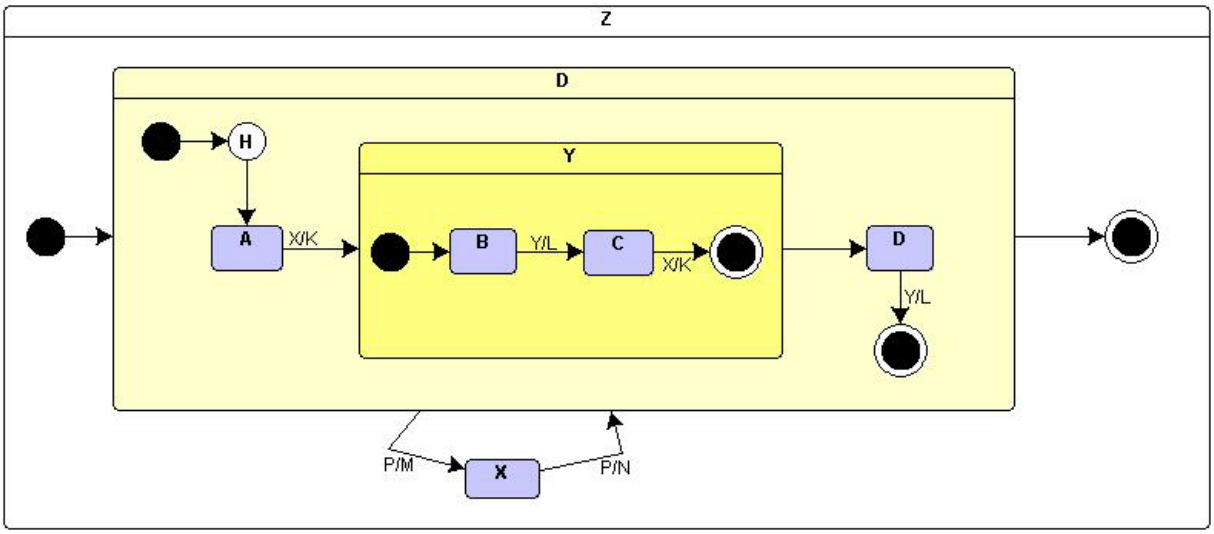
c) (2 Punkte) Aus jedem Zustand innerhalb von **C-2** soll bei Eingabe von **R** in den Zustand **X** gewechselt werden. Vom Zustand **X** soll genau dann in den Zustand **S-1** gewechselt werden, wenn die Bedingung ***setOn*** gilt.

d) (4 Punkte) Bei Eingabe von **T** soll von jedem Zustand innerhalb von **C-1** in den Zustand **Z** gewechselt werden. Vom Zustand **Z** soll bei Eingabe von **T** genau dann in den Zustand **Y** gewechselt werden, wenn der Zustand **Z** von **Y** aus erreicht wurde. In allen anderen Fällen soll vom Zustand **Z** bei Eingabe von **T** in den Zustand **S-2** gewechselt werden.

Punkte: ____ /10

Aufgabe 4

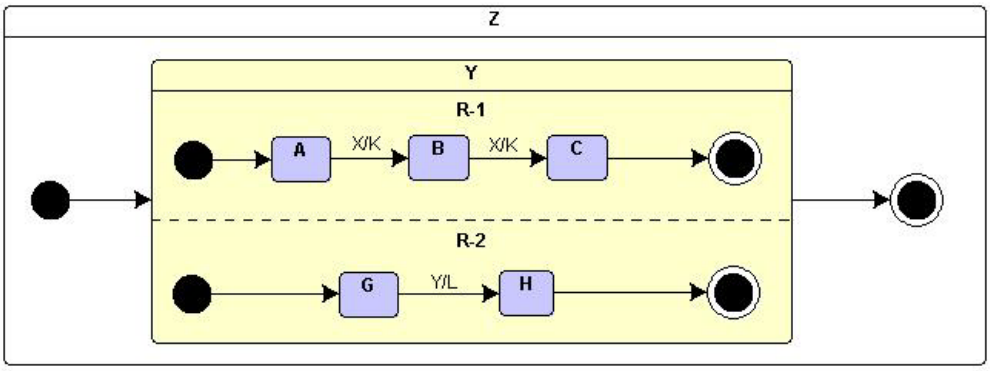
(6 Punkte) Gegeben sei das folgende Zustandsdiagramm. Geben Sie für den Zustand **Z** eine Modellierung an, die das gleiche Ein-/Ausgabe-Verhalten wie **Z** aufweist, aber ganz ohne Historie-Zustände H oder H* auskommt.



Punkte: ____ /6

Aufgabe 5

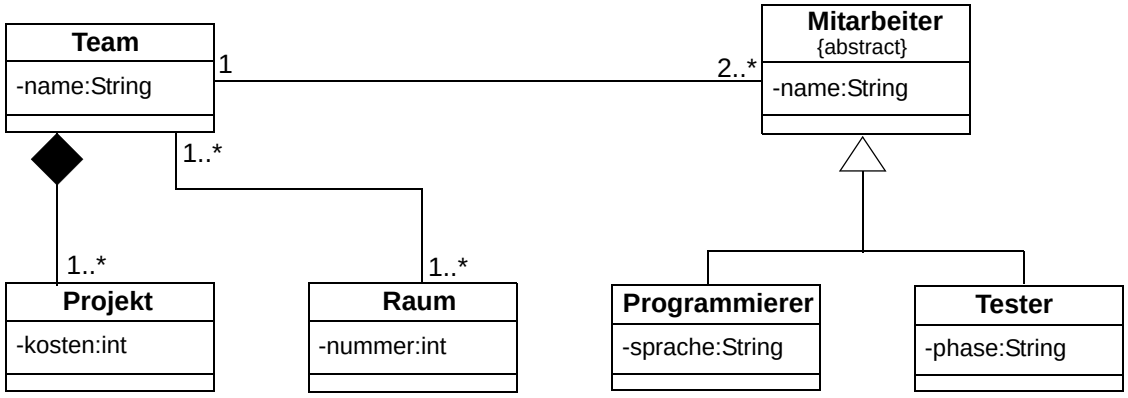
(6 Punkte) Gegeben sei das folgende Zustandsdiagramm. Geben Sie für den Zustand **Z** eine Modellierung an, die das gleiche Ein-/Ausgabe-Verhalten wie **Z** aufweist, aber ganz ohne nebenläufige Zustände auskommt.



Punkte: /6

Aufgabe 6

(6 Punkte) Zeichnen Sie zu dem vorgegebenen Klassendiagramm ein ***zulässiges*** Objektdiagramm, das ***genau zwei*** Objekte der Klasse **Team**, ***insgesamt genau 9*** Objekte umfasst und **alle** Attributwerte zeigt.

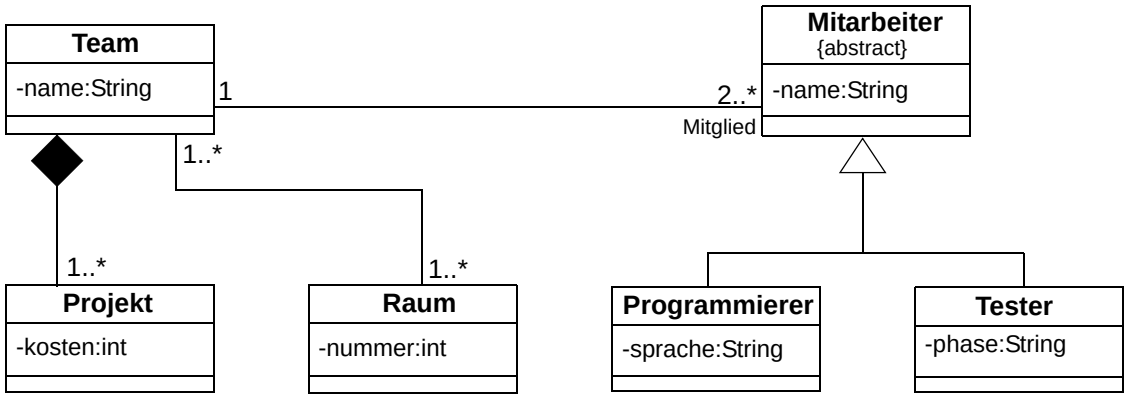


Punkte: ____ /6

Aufgabe 7

(12 Punkte) Erweitern Sie das aus Aufgabe 6 bekannte Klassendiagramm derart, dass es die nachfolgende Beschreibung modelliert. Verwenden Sie die bereits vorgegebenen Elemente, wo dieses möglich ist, oder ergänzen Sie weitere Elemente.

- Jeder Raum ist in einem Gebäude, ein Gebäude kann mehrere Räume besitzen. Ein Gebäude hat genau eine Bezeichnung.
- Jedem Mitarbeiter stehen mehrere Rechner zur Verfügung, die durch einen Namen gekennzeichnet sind. Jeder Mitarbeiter besitzt auf jedem Rechner ein eigenes Passwort. Ein Rechner kann von mehreren Mitarbeitern benutzt werden.
- DB-Programmierer sind besondere Programmierer, die ein Attribut db besitzen, das die Datenbank bezeichnet, die sie programmieren können.
- Jedes Team hat einen Mitarbeiter als Teamleiter, der selbst Mitglied des Teams ist.
- Tester können an Audits teilnehmen. Ein Audit wird von höchstens 4 Testern durchgeführt, jeder Tester kann aber an beliebig vielen Audits teilnehmen. Ein Audit nimmt eine Gesamtbewertung mit einer Note zwischen 1 und 5 für genau ein Projekt vor. Jeder Tester, der an einem Audit teilnimmt, vergibt eine einzelne Note zwischen 1 und 5.



Gebäude	DB-Programmierer	Audit	Rechner

Punkte: ____ /12

Aufgabe 8

(12 Punkte) Erstellen Sie ein Klassendiagramm für die Entwicklung eines Bibliothekssystems (BibSys), das die folgenden Eigenschaften modellieren soll. Verwenden Sie bei Ihrer Modellierung die schon vorhandenen Klasse.

- BibSys verwaltet Nutzer und Medien.
- Jeder Nutzer besitzt einen Namen und eine Nutzernummer.
- Jedes Medium hat einen Titel.
- Spezielle Medien sind Bücher mit einer ISBN-Nummer und Videos mit einer Format-Angabe.
- Jeder Nutzer hat bis zu zehn Interessensgebiete.
- Für jedes Interessensgebiet sind jedem Nutzer eine beliebige Zahl Medien zugeordnet, an denen der Nutzer Interesse und die er daher vorgemerkt hat.
- Zusätzlich kann jeder Nutzer bis zu 10 Medien ausleihen. Für jedes Medium muss dafür ein Rückgabedatum festgehalten werden. Ein Medium kann immer nur an genau einen Nutzer ausgegeben werden.

Datum

Punkte: ____ /12

Aufgabe 9

(12 Punkte) Nehmen Sie an, dass die in der folgenden Methode `show` benutzten Klassen und Methoden ihren Definitionen entsprechend verwendet und aufgerufen werden. Erstellen Sie ein Sequenzdiagramm, das den Ablauf eines Aufrufs der Methode `show` zeigt.

```
public void show(List aList) {
    ListBox aBox = new ListBox();
    int co = aList.size();
    while (co > 0) {
        aBox.add(aList.get(co).getName());
        co--;
    }
}
```

Punkte: ____ /12

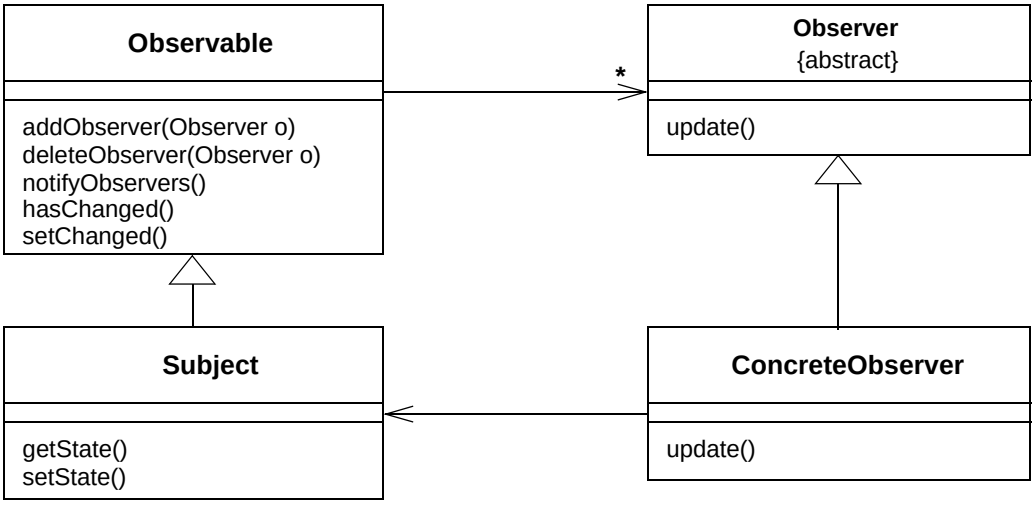
Aufgabe 10

(3 Punkte) Zeichnen Sie ein Aktivitätsdiagramm, welches den Ablauf der aus Aufgabe 9 bekannten Methode `show` visualisiert.

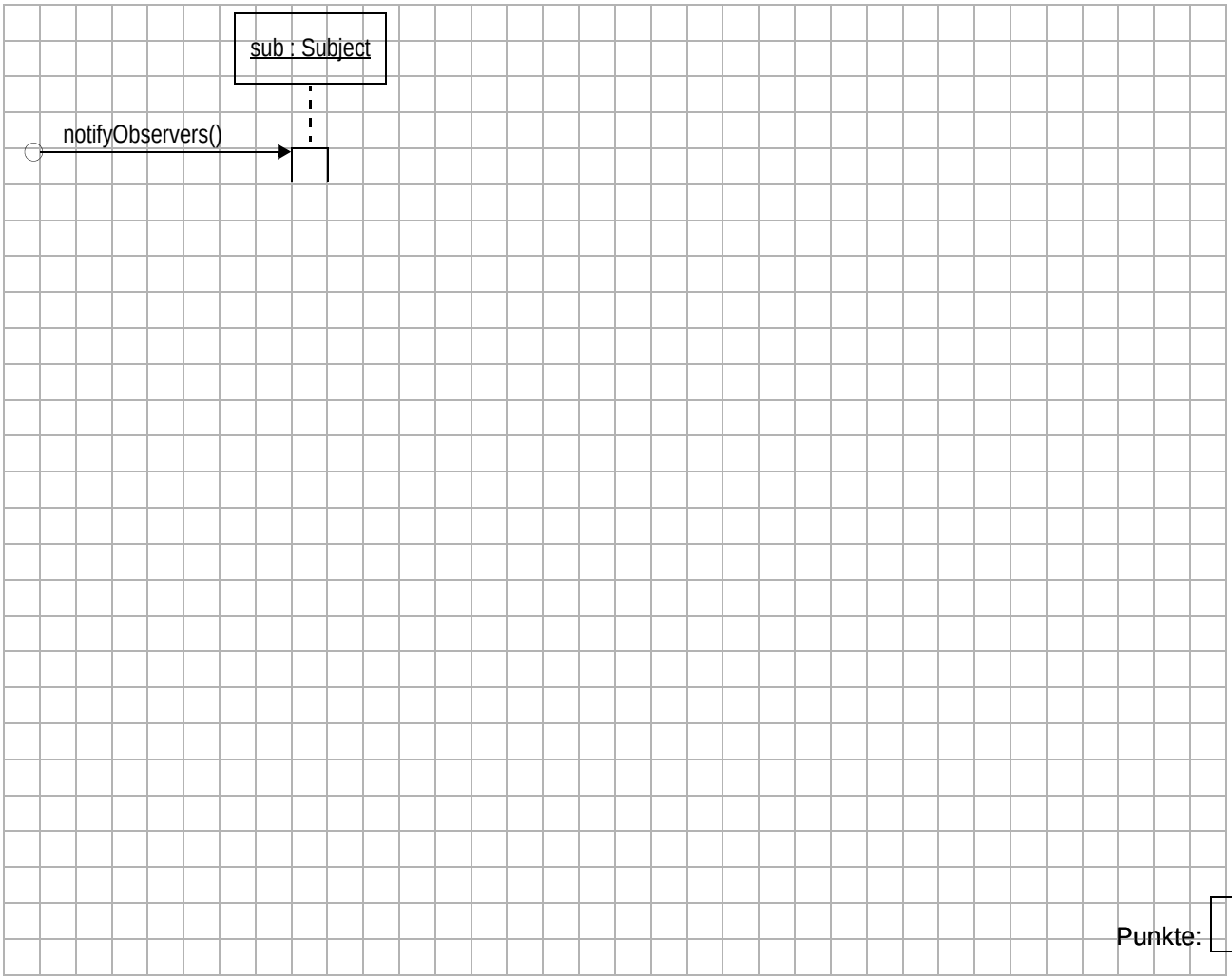
```
public void show(List aList) {
    ListBox aBox = new ListBox();
    int co = aList.size();
    while (co > 0) {
        aBox.add(aList.get(co).getName());
        co--;
    }
}
```

Punkte: ____ /3

Aufgabe 11



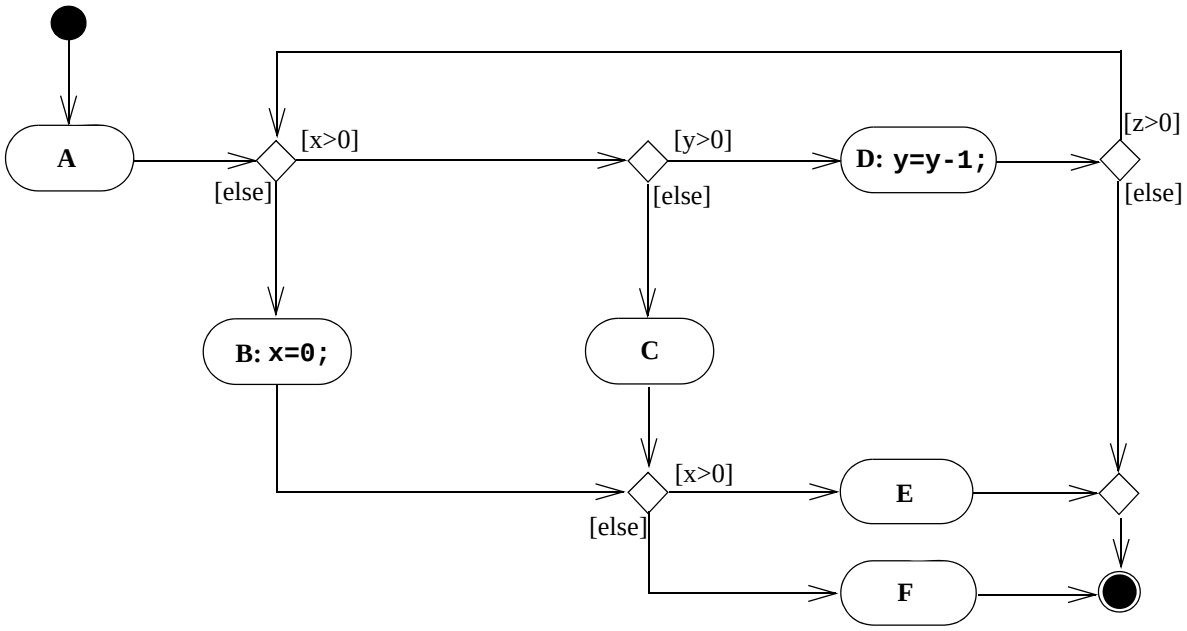
(10 Punkte) Das Klassendiagramm zeigt das aus der Vorlesung bekannte Entwurfsmusters *Beobachter*. Vervollständigen Sie das unten begonnene Sequenzdiagramm derart, dass es für einen Aufruf der Methode `notifyObservers()` die Aufrufe der verschiedenen im Klassendiagramm vorgegeben Methoden zeigt. Gehen Sie davon aus, dass *n* Objekte der Klasse `ConcreteObserver` bereits alle bei `sub` angemeldet sind.



Punkte: ____ /10

Aufgabe 12

Gegeben ist das folgende Aktivitätsdiagramm:



Alle Aktionen enthalten neben neben den beiden angeführten Zuweisungen (in **B** und **D**) weitere Operationen, die beim Testen ausgeführt werden sollen, im Detail aber nicht wichtig und daher im Diagramm nicht aufgeführt sind. Insbesondere werden die Werte der Variablen `x` und `y` nur durch die beiden angegebenen Zuweisungen geändert, der Wert von `z` wird nicht verändert.

Bestimmen Sie Testfälle. Geben Sie jeweils eine **minimale** Anzahl von Testfällen als Wertebelegungen für die `int`-Variablen x, y und z an, so dass die folgenden Überdeckungen den **maximal** möglichen Wert erreichen.

a) (2 Punkte) *vollständige Anweisungsüberdeckung*:

b) (2 Punkte) *vollständige Zweigüberdeckung*:

c) (2 Punkte) *vollständige strukturierte Pfadüberdeckung mit k=1*:

d) (2 Punkte) *vollständige strukturierte Pfadüberdeckung mit k=2*:

Punkte: ____ /8