VU Objektorientierte Modellierung	**Übung 4: Zustandsdiagramm**
	Übungsgruppen: 15.5.2008 - 21.5.2008

Aufgabe 1: Theoriefragen 1

Beantworten Sie folgende Fragen:

a) Erklären Sie die Konzepte *Ereignis*, *Bedingung* und *Aktivität*.

b) Welche Art von Aktivitäten gibt es innerhalb eines Zustands?

c) Wann erfolgt eine Transition (von einem Zustand in einen anderen)?

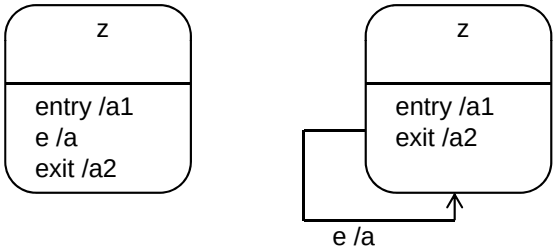
d) Was versteht man unter einem Historischen Zustand? Wann, warum und wie wird er eingesetzt?

Aufgabe 2: Theoriefragen 2

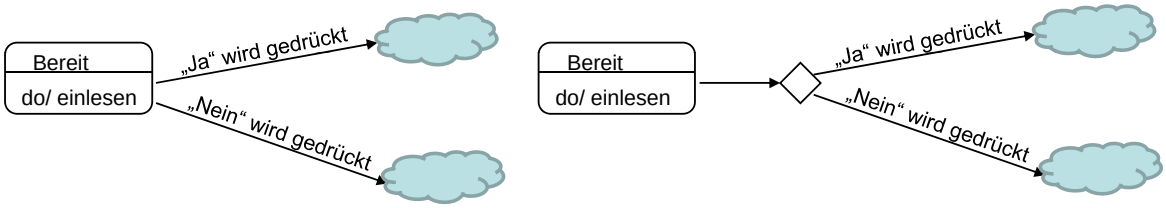
Beantworten Sie folgende Fragen:

a) Erklären Sie das Konzept der UND- sowie der ODER-Verfeinerung.

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



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

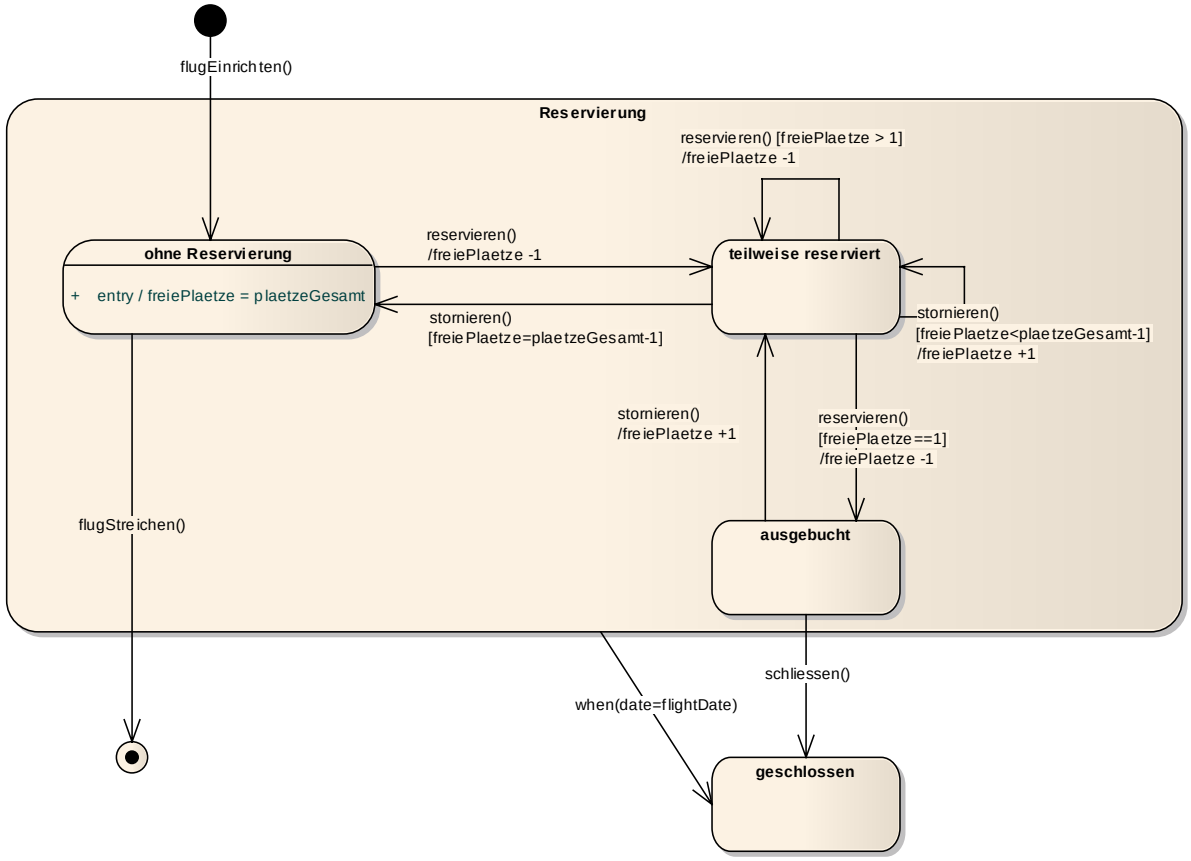


d) Gegeben sind folgende zwei Ausschnitte eines Zustandsdiagramms. Sind die beiden Ausschnitte äquivalent? Begründen Sie Ihre Antwort!

Aufgabe 3: Flugreservierung

Modellieren Sie ein UML 2.0 Zustandsdiagramm, das die Zustände von Sitzplatzreservierungen eines Fluges (aus der Sicht der Fluggesellschaft) abbildet.
Es stehen folgende Operationen zur Verfügung:

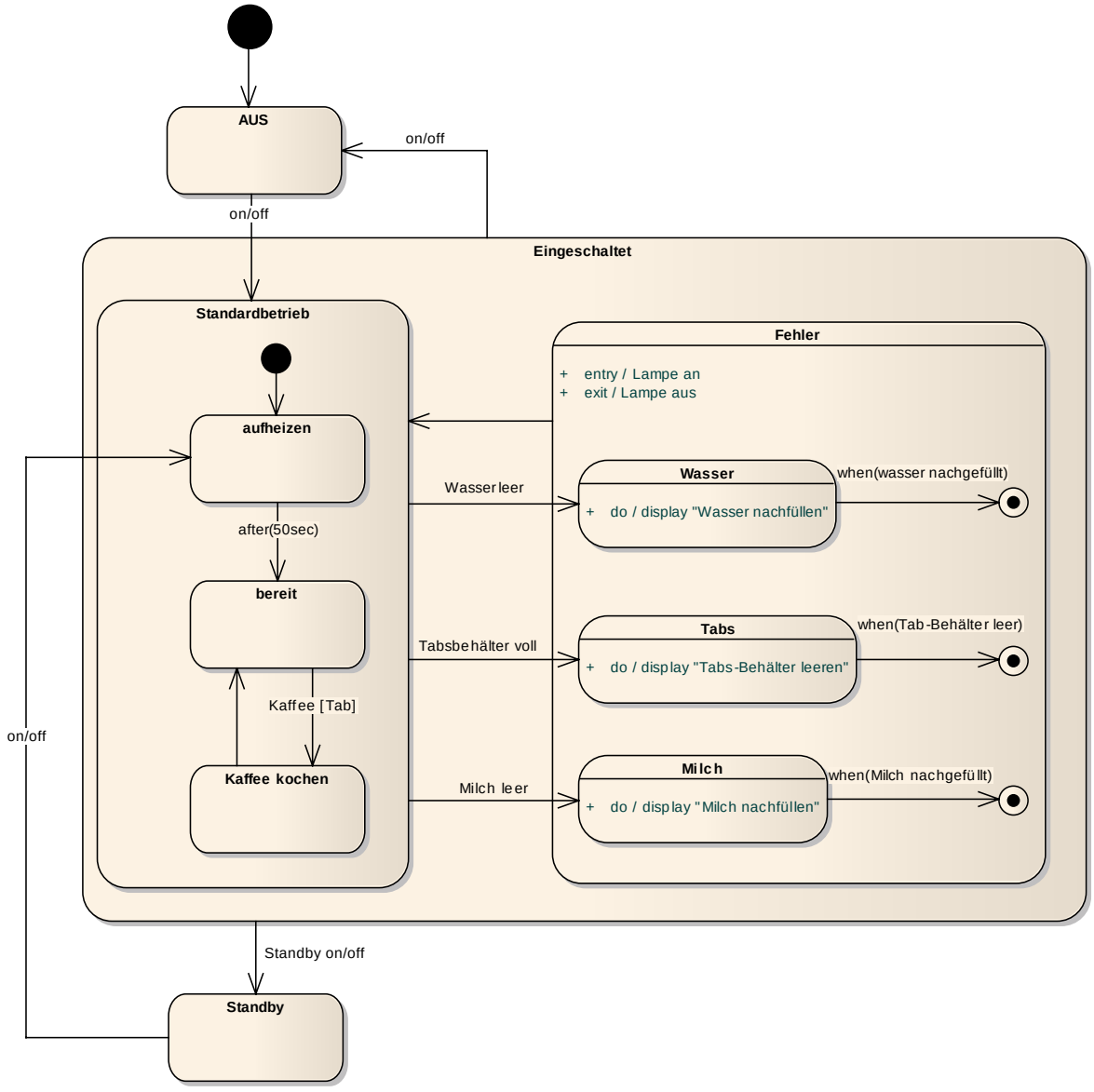
- `flugEinrichten()`: Ein neuer Flug wird „eröffnet“; freiePlaetze=PlaetzeGesamt
- `reservieren()`: Wird diese Operation zum ersten Mal aufgerufen, wechselt das System in den Zustand „teilweise reserviert“; anschließend ist es so lange möglich, weitere Sitzplätze zu reservieren, bis der Flug ausgebucht ist.
- `stornieren()`: Ein reservierter Sitzplatz wird wieder frei
- `schliessen()`: Ein Flug kann geschlossen werden, sobald dieser ausgebucht ist, oder das Abflugdatum das aktuelle Datum ist
- `flugStreichen()`: ein Flug kann nur dann gestrichen werden, wenn es noch keine reservierten Plätze gibt.



Aufgabe 4: Kaffeemaschine

Modellieren Sie die Zustände, die eine (vereinfachte) Nespresso-Kaffeemaschine einnehmen kann, mittels UML 2.0-Zustandsdiagramm. Die Kaffeemaschine besitzt eine digitale Anzeige, eine rote LED-Lampe, sowie 3 Knöpfe (`On/Off`, `Standby On/Off`, `Kaffee`). Zu Beginn befindet sich die Kaffeemaschine (nach Betätigung des On/Off-Knopfes) im Zustand „Aufheizen“. Nach 50 Sekunden geht die Kaffeemaschine in den Zustand „Bereit“ über und verweilt in diesem so lange, bis der On/Off oder der Standby On/Off - Knopf betätigt wird.
Wird die Taste `Kaffee` gedrückt, wird ein Kaffeee gemacht, sofern davor ein Kaffee-Tab in die Maschine eingelegt wurde.
Sollte es irgendein Problem geben, so wechselt die Maschine in den Zustand `Fehler` und verharrt darin so lange, bis der Fehler behoben wurde. Am Display der Kaffeemaschine wird eine konkrete Fehlermeldung mit einer Handlungsanweisung angezeigt (zB: „Kapselbehälter leeren“). Die rote LED-Lampe leuchtet immer dann auf, wenn aufgrund eines Problems gerade kein Kaffee gemacht werden kann; das Licht erlischt erst dann, wenn das jeweilige Problem behoben wurde. Überlegen und modellieren Sie mind. drei sinnvolle Problemfälle (inklusive Fehlermeldungen etc.).
Durch Drücken des Knopfes „Standby On/Off“ wird die Kaffeemaschine in den Standby-Zustand versetzt, egal in welchem Zustand sie sich vorher befunden hat. Vom Standby-Zustand geht die Maschine stets in den Zustand „Aufheizen“ über.

Annahme: Die Maschine muss nach einem Fehler immer aufheizen



Aufgabe 5: Zustandsdiagramm mit Call-Events

Gegeben sei das auf der folgenden Seite dargestellte Klassendiagramm, das einen Ausschnitt eines Informationssystems zur Zeugnisverwaltung darstellt:

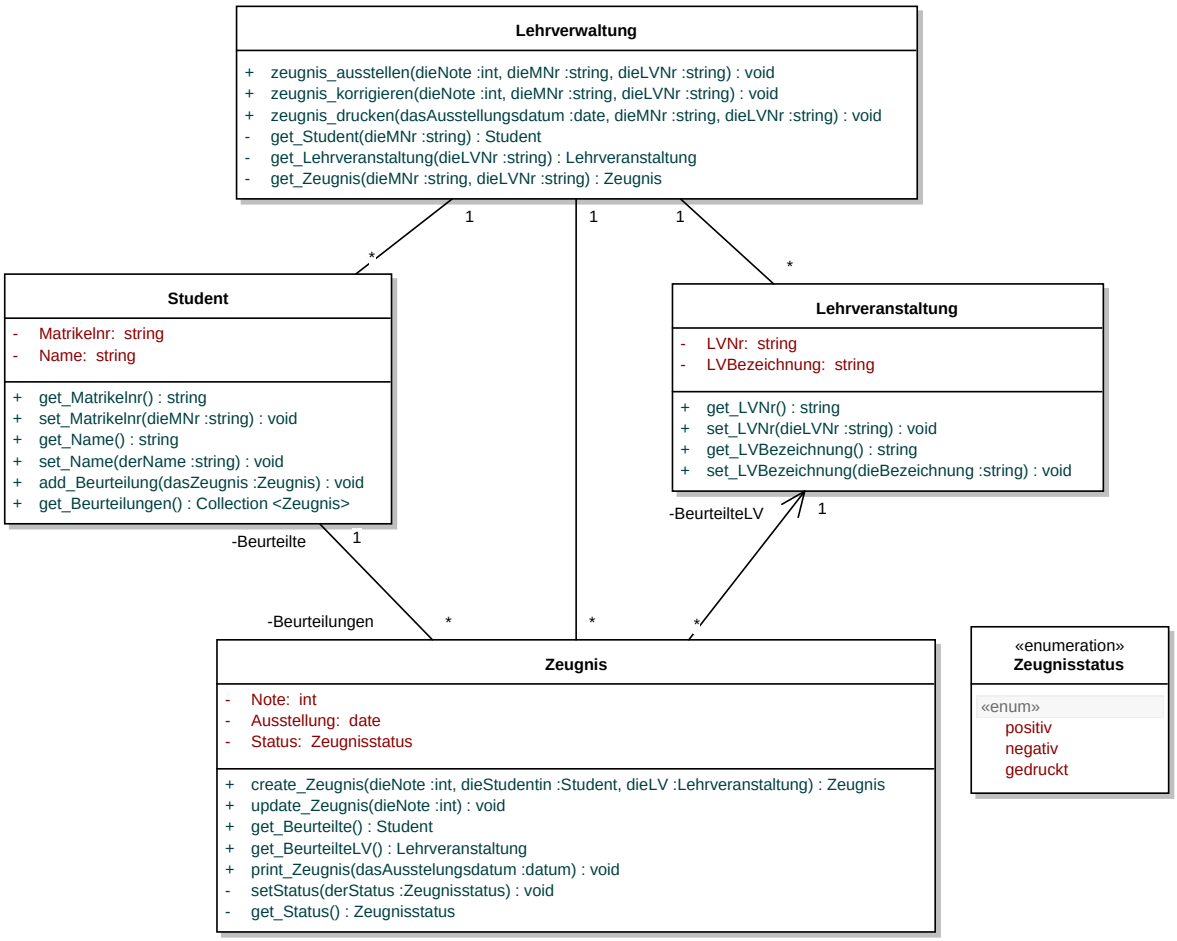
Die Klasse `Lehrverwaltung` bildet die Geschäftslogik ab. Sie verwaltet jeweils mehrere Objekte der Klassen `Student`, `Lehrveranstaltung` und `Zeugnis`. Ein `Zeugnis` bezieht sich auf genau einen `Studenten` (der `Beurteilte`) und eine `Lehrveranstaltung` (`die BeurteilteLV`). Ein `Student` erhält auch mehrere `Zeugnisse` (die `Beurteilungen`). Für eine `Lehrveranstaltung` werden zwar mehrere `Zeugnisse` ausgestellt, doch diese Beziehung ist nicht navigierbar. Das heißt, die Menge der `Zeugnisse`, die für eine `Lehrveranstaltung` ausgestellt wurden, werden NICHT in einer Collection verwaltet.

Eine `Lehrveranstaltung` umfasst die eindeutig identifizierende Lehrveranstaltungsnummer (`LVNr`) und die Bezeichnung der Lehrveranstaltung (`LVBezeichnung`). Get- und Set-Operationen zum Ermitteln bzw. Setzen dieser Attribute sind vorhanden.

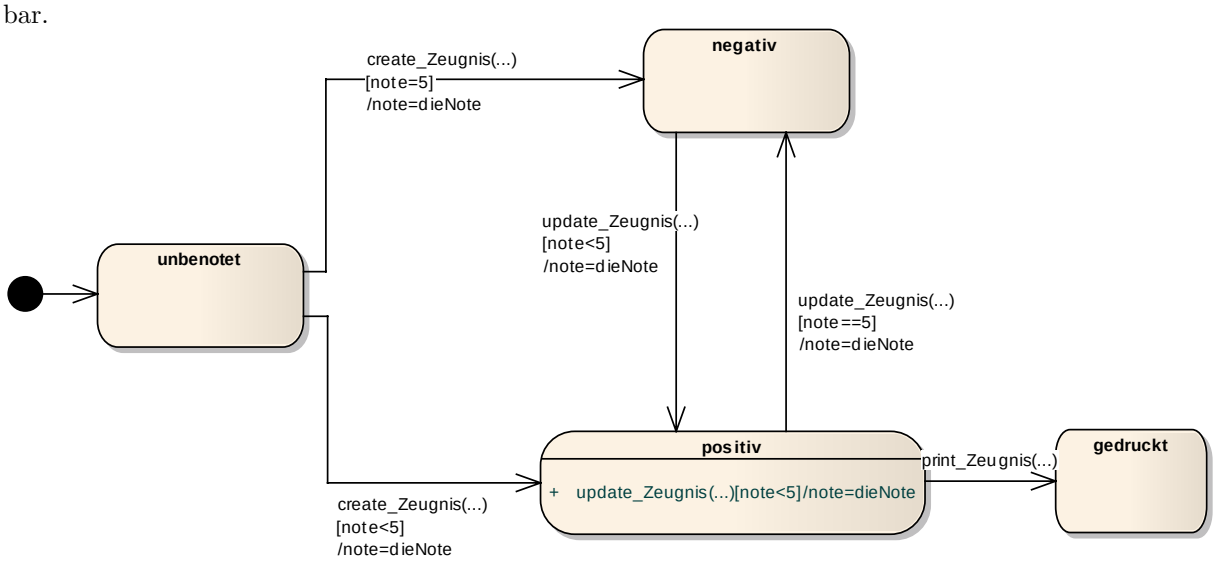
Ein `Student` ist durch die eindeutig identifizierende Matrikelnummer (`Matrikelnr`) und den `Namen` gekennzeichnet. Wieder gibt es Get- und Set-Operationen für diese Attribute. Zusätzlich gibt es noch Operationen für die Beziehung zur Klasse `Zeugnis`. Die Operation `add_Beurteilung` fügt ein `Zeugnis` hinzu, und die Operation `get_Beurteilung` ermittelt die für den `Studenten` ausgestellten `Zeugnisse`.

Die Klasse `Zeugnis` hat die Attribute `Note`, das Datum der Ausstellung (`Ausstellung`), und den `Status`. Der `Status` ist einer der folgenden Werte, die im Aufzählungstyp `Zeugnisstatus` definiert sind: `positiv`, `negativ` oder `gedruckt`. Ein `Zeugnis` wird durch die public Operation `create_Zeugnis` erzeugt, welche als Input jeweils ein Objekt der Klasse `Zeugnis` und der Klasse `Lehrveranstaltung` bzw. einen Integer-Wert für die `Note` erwartet. Die Note eines Zeugnis kann durch *update_Zeugnis* akualisiert werden. Die Operation *print_Zeugnis* führt zur Ausstellung an einem gewissen Datum. Für die `Note` und das Datum der `Ausstellung` gibt es weiters Get-Operationen. Für den `Status` private Get- und Set-Operationen.

Die Klasse `Lehrveranstaltung` bildet alleine die Geschäftslogik ab. Sie stellt der Anwendung die selbst-erklärenden Operationen `zeugnis_ausstellen`, `zeugnis_korrigieren` und `zeugnis_drucken` zur Verfügung. An diese Operationen werden die Strings für die Matrikelnummer und die Lerveranstaltung neben Note bzw. Ausstelungsdatum übergeben. Die Klasse `Lehrveranstaltung` enthält noch private Operationen zum Ermitteln von Objekten der Klasse `Student`, `Lehrveranstaltung` bzw. `Zeugnis` auf Grund der als String übergebenen Marikelnummer und/oder Lehrveranstaltungsnummer.

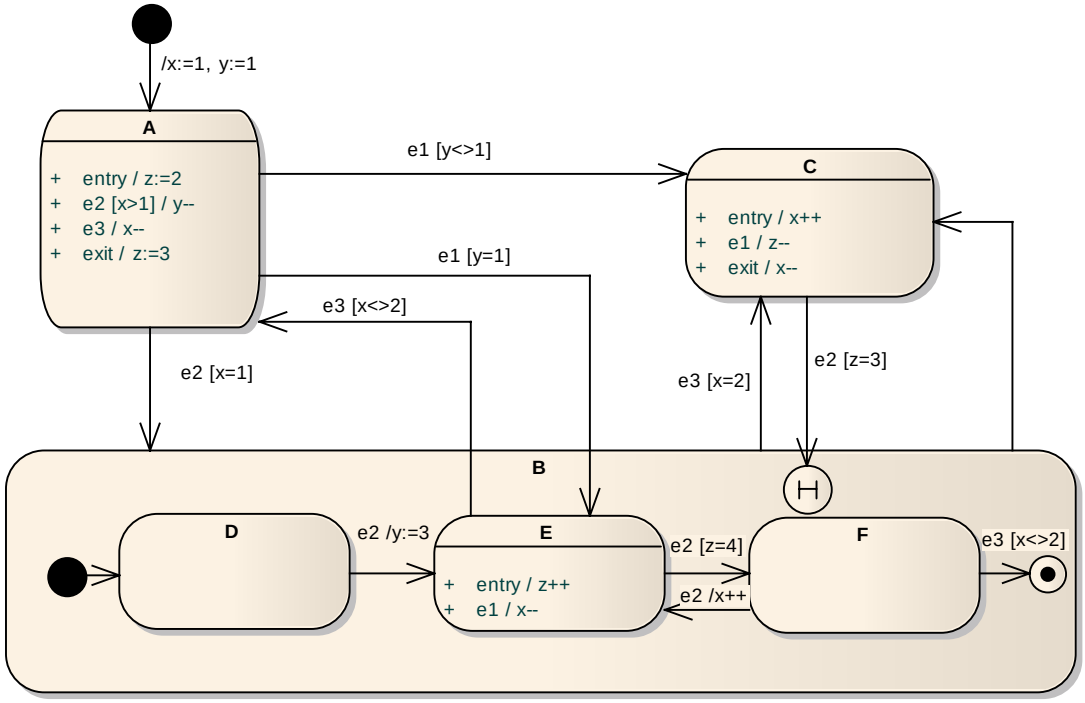


Vervollständigen Sie das gegebene Zustandsdiagramm, um die Zustände von `Status` der Klasse `Zeugnis` zu modellieren. Die Zustandsübergänge werden durch Call-Events der entsprechenden public Operationen ausgelöst. In dem zu entwickelnden studentenfreundlichen System werden Zeugnisse, die mit Nicht Genügend (5) beurteilt wurden, nicht ausgedruckt. Einmal ausgedruckte Zeugnisse sind nicht mehr änderbar.



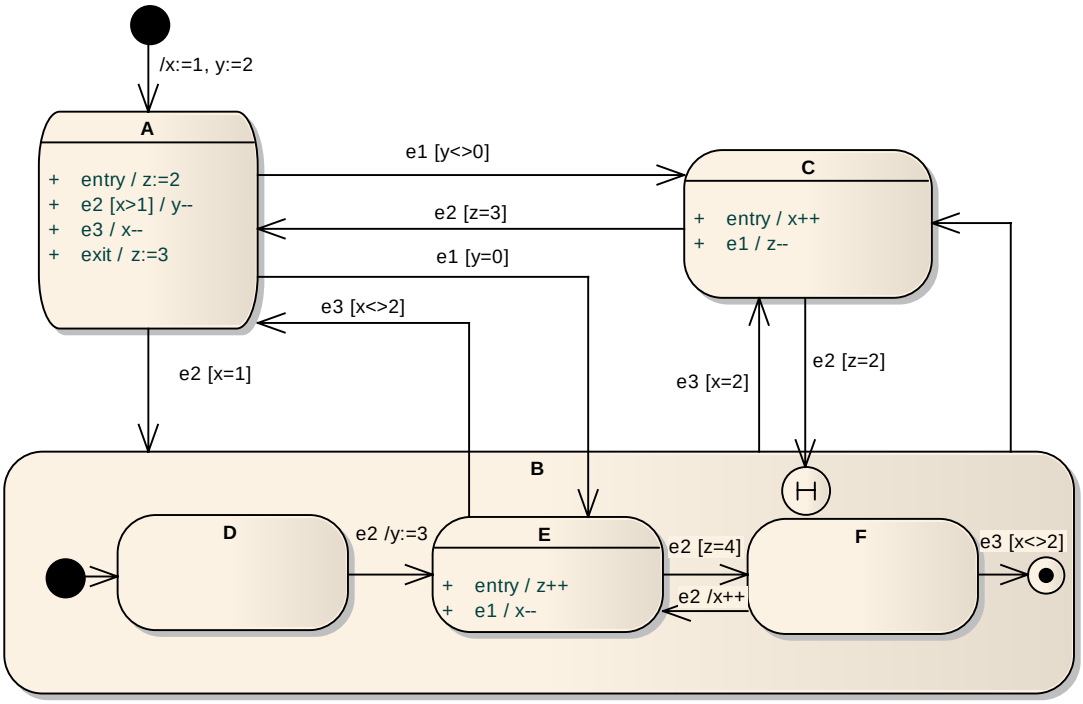
Aufgabe 6: Ereignisfolgen

a) Gegeben ist das nachfolgende Zustandsdiagramm. Welche Zustände und Aktionen kommen bei der Ereignisfolge e1 e2 e3 e1 e2 e3 vor? Entwerfen Sie hierzu eine Tabelle mit den 3 Spalten *Ereignis*, *eingetretener Zustand* und *durchgeführte Aktion(en)*.



		Variablenwerte		
Ereignis	Eingetr. Zustand	x	y	z
Beginn	A	x=1	y=1	z=2
e1	B/E	–	–	z=3, z=4
e2	B/F	–	–	–
e3	C	x=2	–	–
e1	C	–	–	z=3
e2	B/D	x=1	–	–
e3	B/D	–	–	–

b) Gegeben ist das nachfolgende Zustandsdiagramm. Welche Zustände und Aktionen kommen bei der Ereignisfolge e2 e2 e3 e1 e1 e2 vor? Entwerfen Sie hierzu eine Tabelle mit den 3 Spalten *Ereignis*, *eingetretener Zustand* und *durchgeführte Aktion(en)*.



		Durchgeführte Aktion(en)		
Ereignis	Eingetr. Zustand	x	y	z
Beginn	A	x=1	y=2	z=2
e2	B/D	–	–	z=3
e2	B/E	–	y=3	z=4
e3	A	–	–	z=2
e1	C	x=2	–	z=3
e1	C	–	–	z=2
e2	E	–	–	z=3