

# Aufgaben zur Modellierung mit UML

---

*Hinweise zur Abgabe:*

Die Abgabe erfolgt als **ZIP-Archiv** in einer Datei **loesung.zip**. Das ZIP-Archiv soll folgende Dateien enthalten: Eine Datei **loesung.txt** mit Antworten zu den Fragen in den Aufgaben 1, 2 und 3 sowie eine Datei **zähler.mdzip** mit den entsprechenden UML-Diagrammen zu Aufgabe 4. Die Abgabe muss **bis 20.11. (09:00 Uhr)** in **1–3er Gruppen** über **GOYA** erfolgen.

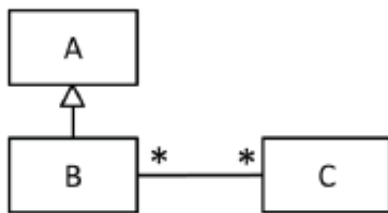
# Aufgabe 1

Welche der folgenden Objektdiagramme passen zu dem abgebildeten Klassendiagramm?

a)

---

Klassendiagramm:



Objektdiagramme:

1. 

```
graph LR
    a1["a1 : A"] --- b1["b1 : B"]
```
2. 

```
graph LR
    a1["a1 : A"] --- c1["c1 : C"]
```
3. 

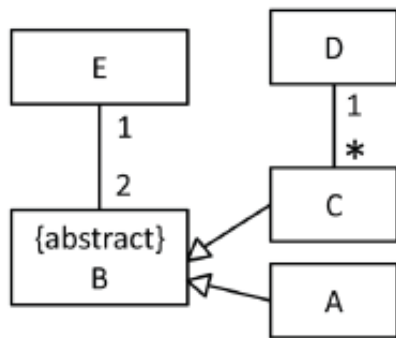
```
graph LR
    b1["b1 : B"] --- c1["c1 : C"]
```
4. 

```
graph LR
    b1["b1 : B"] === c1["c1 : C"]
```
5. 

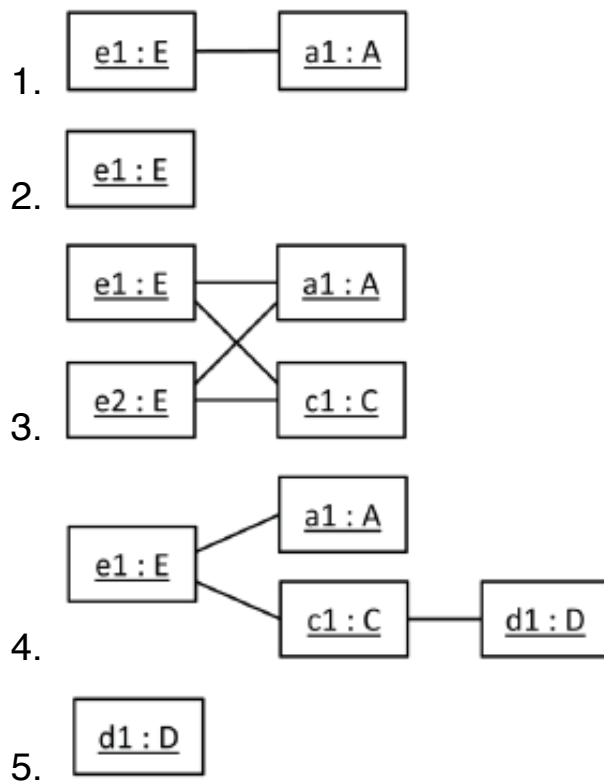
```
graph LR
    c1["c1 : C"]
```

b)

Klassendiagramm:



Objektdiagramme:




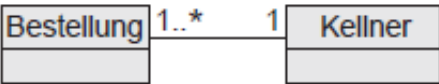

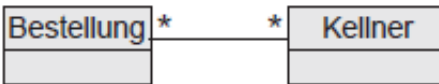
## Aufgabe 2

Wie sollte jeweils der folgende Sachverhalt mit einem UML-Klassendiagramm modelliert werden?

a)

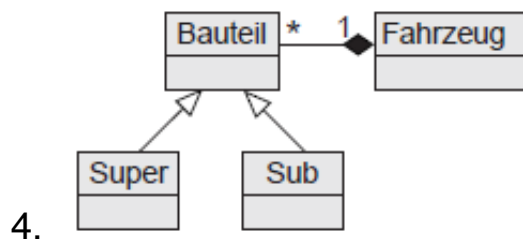
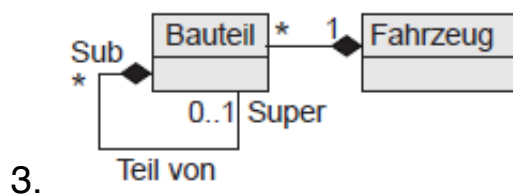
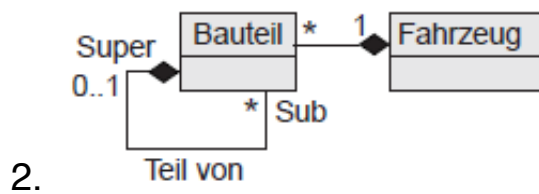
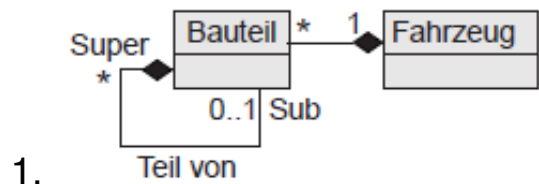
---

Eine Bestellung wird bei genau einem Kellner aufgegeben, ein Kellner kümmert sich um mehrere (0 oder mehr) Bestellungen.

1. 
2. 
3. 
4. 

b)

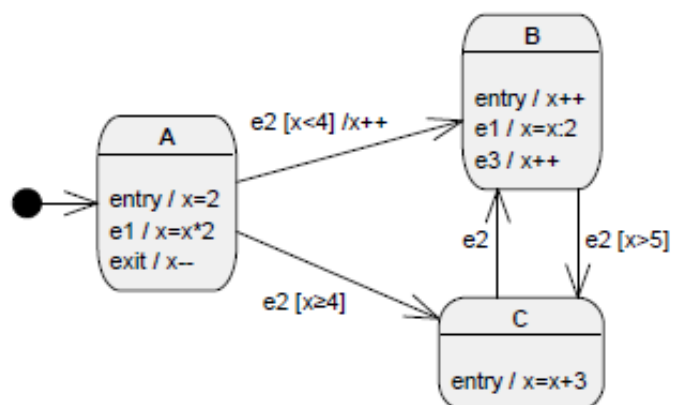
Ein Fahrzeug besteht aus einer Vielzahl an Bauteilen. Jedes Bauteil kann wiederum in maximal einem übergeordneten Bauteil (Superbauteil, "Super") eingebaut sein, ein Superbauteil setzt sich aus mehreren untergeordneten Bauteilen (Subbauteile, "Sub") zusammen.



## Aufgabe 3 (5+5 Punkte)

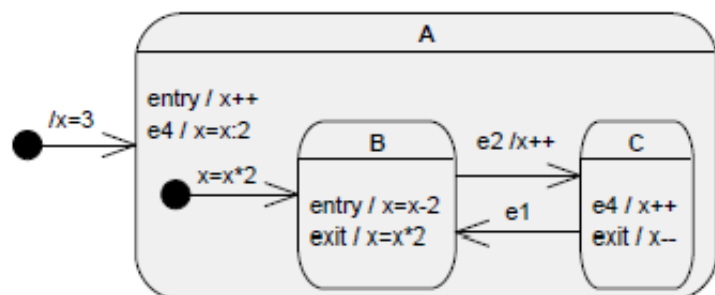
### Teil a)

Betrachten Sie nachfolgenden Zustandsautomaten. Welchen Wert hat  $x$  nach der Ereignisfolge  $e_2 e_3 e_1 e_2 e_3$ ?



### Teil b)

Betrachten Sie nachfolgenden Zustandsautomaten. Welchen Wert hat  $x$  nach der Ereignisfolge  $e_4 e_2 e_4 e_1$ ?



## Aufgabe 4 (10 Punkte)

Modellieren Sie das Verhalten eines Sekundenzählers einer Digitaluhr mit Hilfe von UML-Zustandsautomaten. Erstellen Sie dazu eine Klasse "Sekundenzähler" mit einem Attribut "zähler" vom Typ "Integer". Die Zeiteinheit der Modellzeit sei 1 Sekunde. Der Zustandsautomat soll den Zähler am Anfang auf 0 setzen. Danach erhöht der Zähler nach jeder Sekunde den Wert seines Zählerattributes und springt nach 60 Sekunden zurück auf 0 (d.h. die Werte liegen zwischen 0 und 59). Zusätzlich soll es möglich sein den Zähler von außen durch ein "Reset"-Signal auf 0 zurückzusetzen, durch ein "Pause"-Signal anzuhalten und durch ein "Resume"-Signal das Zählen fortzusetzen.

### Modellierungshinweise

---

- Für die Beschreibung von Aktionen und Bedingungen können Anweisungen und Ausdrücke aus einer imperativen Programmiersprache eingesetzt werden. Es kann aber auch natürliche Sprache benutzt werden.