

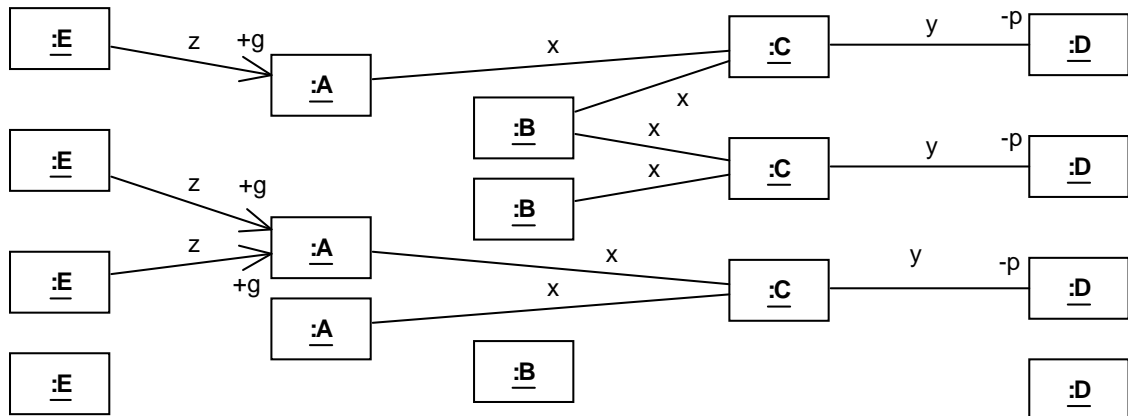
**Hinweise zu den Modellierungsbeispielen:**

- Bilden Sie den Sachverhalt, der in der Angabe geschildert wird, moeglichst genau ab.
- Datentypen muessen Sie nur darstellen, wenn diese explizit im Text/Code spezifiziert wurden.
- Sollte etwas in der Angabe nicht erwaehnt sein, treffen Sie sinnvolle Annahmen.

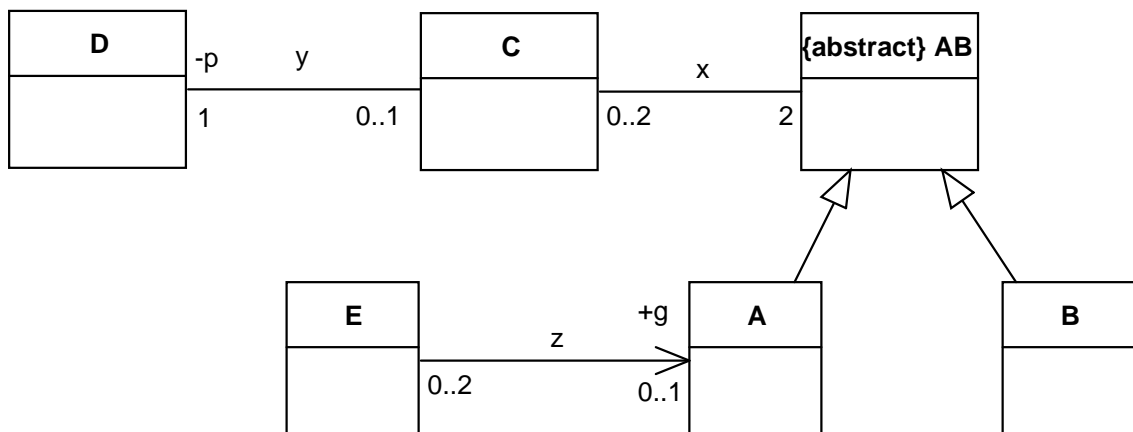
**Aufgabe 1: Objektdiagramm**

Entwerfen Sie zwei Klassendiagramme, zu denen nachfolgende Objektdiagramme konform sind. Wählen Sie die Kardinalitäten an den Assoziationsenden möglichst genau. Sie können davon ausgehen, dass diese Objektdiagramme die minimal und die maximal mögliche Anzahl an Beziehungen mit Objekten einer anderen Klasse darstellen. Der Name jeder Beziehung ist im Klassendiagramm eindeutig (es sollen also keine Beziehungen denselben Namen haben). Weiters sollen mögliche Generalisierungen bzw. XOR-Beziehungen erkannt werden.

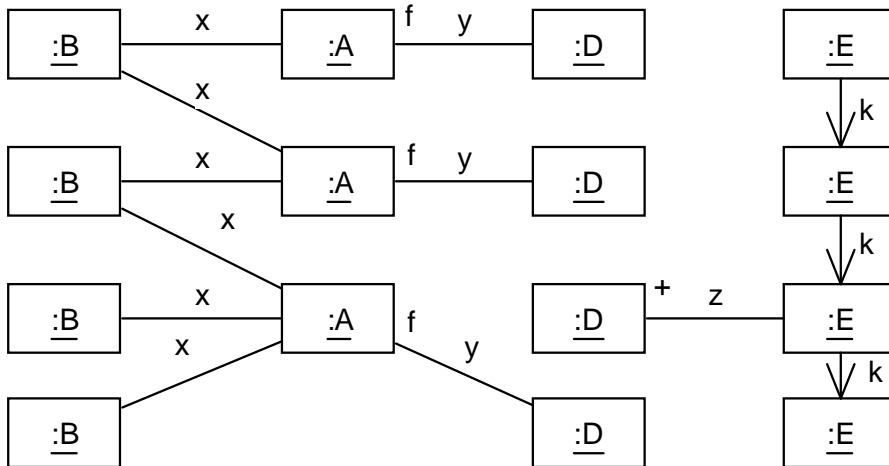
• **Objektdiagramm 1:**



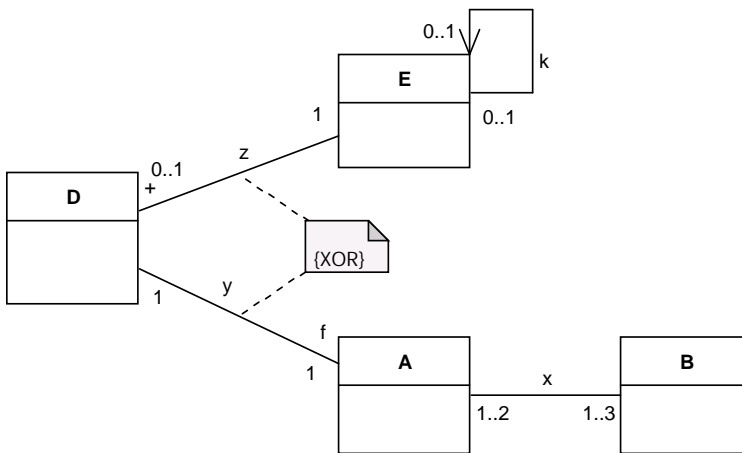
**Klassendiagramm 1:**



• Objektdiagramm 2:



Klassendiagramm 2:



## Aufgabe 2: Vergleich von Klassendiagrammausschnitten

Erklären Sie den Unterschied zwischen folgenden Klassendiagrammausschnitten:

a)



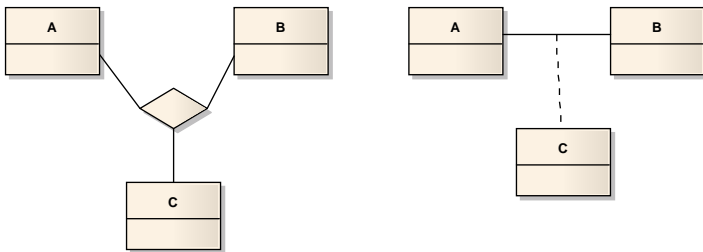
b)



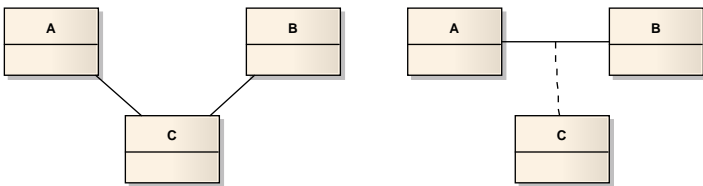
c)



d)



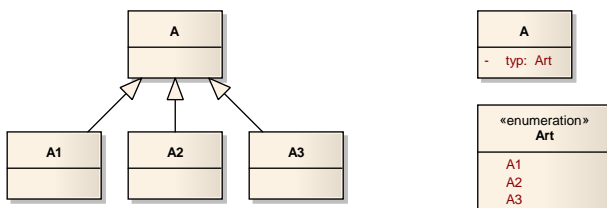
e)



f)



g)



### Aufgabe 3: Reverse Engineering

Gegeben sei der unten angeführte Java ähnliche Code. Führen Sie ein Reverse Engineering des Codes in ein UML Klassendiagramm durch. Das heißt, Sie müssen ein UML Klassendiagramm entwerfen, das semantisch dem Java Code entspricht. Bilden Sie Referenzen möglichst durch Assoziationen ab.

```
abstract class Haus {
    public String name;
    private Art bauart;
    private Besitzer hb;
}
```

```
class Ferienhaus extends Haus {
    public String adresse;
    public Besitzer [] mieter;
}
```

```
enum Art {
    holz ,
    beton ,
    ziegel;
}
```

```
enum Typ {
    privat ,
    firma;
}
```

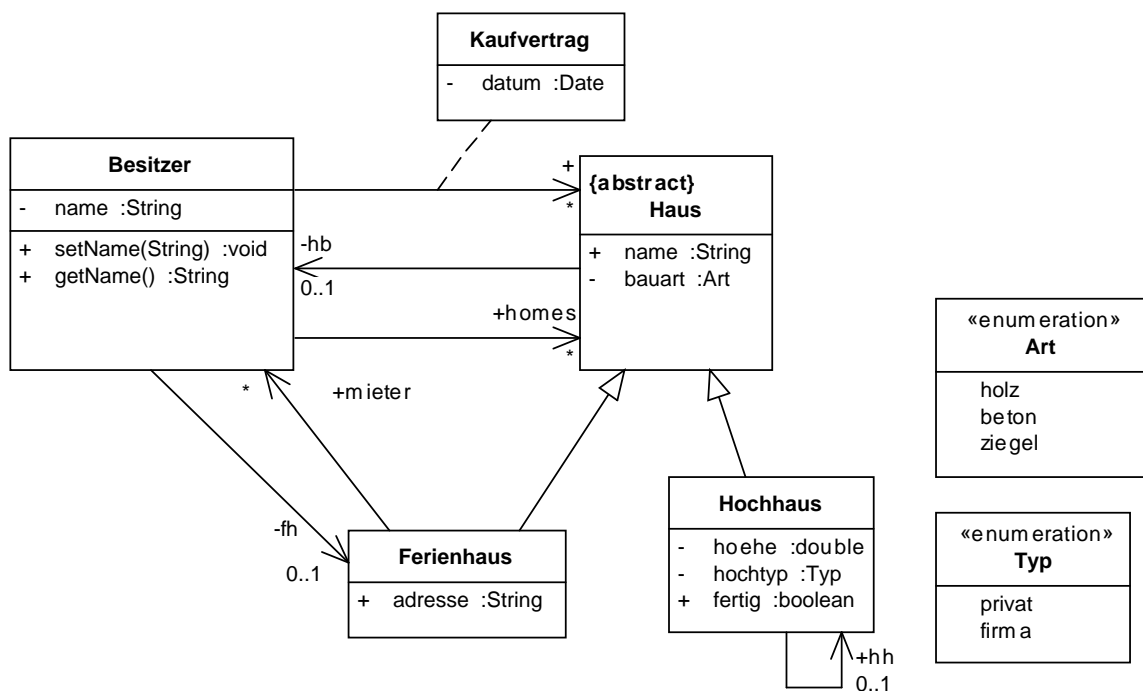
```
class Hochhaus extends Haus {
    private double hoehe;
    private Typ hochtyp;
    public boolean fertig;
    public Hochhaus hh;
}
```

```
class Besitzer {
    private String name;
    private Ferienhaus fh;
    public Haus [] homes;

    public Hashtable kaufvertrag;
    // Key: Haus (Typ: Haus)
    // Value: datum (Typ: Date)

    public String getName() {
        return name;
    }

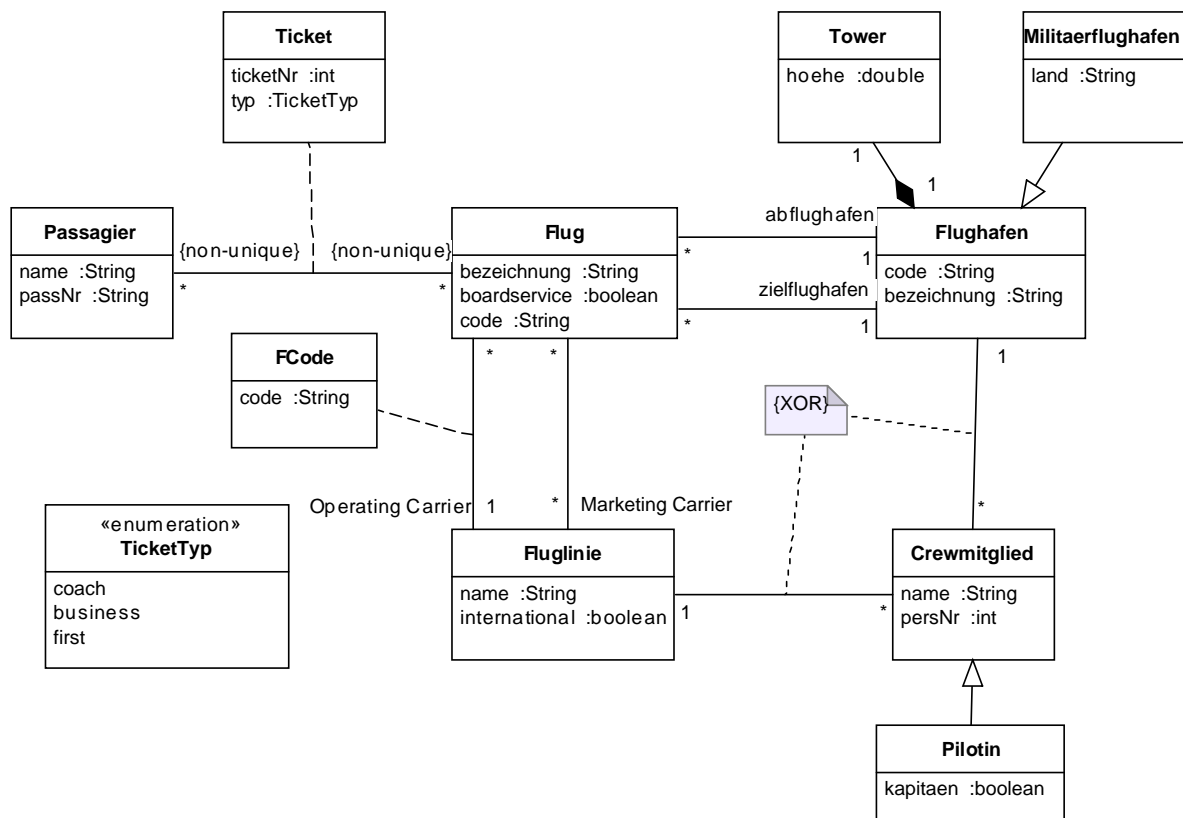
    public void setName(name) {
        this.name = name;
    }
}
```



## Aufgabe 4: Flug

Bilden Sie die folgenden Sachverhalte mit einem Klassendiagramm ab.

Von jedem Flughafen werden seine Bezeichnung und sein Code gespeichert. Von Militärflughäfen wird zusätzlich das Land gespeichert, zu dem sie gehören. Jeder Flughafen hat einen Tower, der fixer Bestandteil dieses Flughafens ist. Von diesem wird die Höhe gespeichert. Von jedem Flug werden die Bezeichnung gespeichert und ob auf diesem ein Boardservice angeboten wird. Jeder Flug hat einen Flughafen als Abflughafen und einen als Zielflughafen. Ein Flug wird grundsätzlich von einer Fluglinie durchgeführt – von denen der Name und ob sie auch internationale Flüge im Programm haben gespeichert wird – es gibt aber auch Flüge mit Codesharing<sup>1</sup>. Jedes Crewmitglied ist entweder bei einem Flughafen oder direkt bei einer Fluglinie angestellt. Von Crewmitgliedern werden Name und Personalnummer gespeichert. Die Pilotin ist ein spezielles Crewmitglied. Von ihr wird gespeichert, ob sie Kapitän ist oder nicht. An jedem Flug nehmen mehrere Passagiere teil die wiederum mehrere Flüge machen können. Von Passagieren werden Name und Passnummer gespeichert. Für jeden Flug den ein Passagier macht gibt es ein Ticket mit einer Ticketnummer und einem bestimmten Tickettyp – coach, business oder first.



<sup>1</sup><https://de.wikipedia.org/wiki/Codesharing>

## Aufgabe 5: Event I

Bilden Sie die folgenden Sachverhalte mit einem Klassendiagramm ab.

Es gibt zwei verschiedene Arten von Veranstaltungen, und zwar Konferenzen und Jubiläen. Von jeder Veranstaltung wird die Bezeichnung gespeichert, von Konferenzen zusätzlich ob eine Förderung bewilligt wurde oder nicht, und von Jubiläen um welche Art Jubiläum es sich handelt - 10, 25 oder 50 Jahre. Ein Eventplaner, dessen Name gespeichert wird, reserviert für eine Veranstaltung Zimmer in mehreren Hotels. Ein Eventplaner kann in einem Hotel Zimmer für mehrere Veranstaltungen reservieren, aber für eine Veranstaltung werden in einem Hotel stets nur von einem Eventplaner Zimmer reserviert. Von jedem Hotel wird die Art der angebotenen Verpflegung gespeichert. Möglich sind dabei Vollpension, Halbpension oder Übernachtung mit Frühstück.

## Aufgabe 6: Event II

Erweitern Sie das Klassendiagramm aus Aufgabe 5 wie folgt:

Für eine Veranstaltung können mehrere Veranstaltungsorte reserviert werden, von denen die Adresse gespeichert wird. Von jeder Reservierung werden die Reservierungsnummer, das Datum für das reserviert wurde sowie der verhandelte Preis gespeichert. Ein Veranstaltungsort beinhaltet entweder bis zu drei Räume für das Galadinner (es wird jeweils der Name des Raumes gespeichert und ob er eine Bühne besitzt) oder mehrere Seminarräume (Bezeichnung und Anzahl der Sitzplätze wird gespeichert). Ein Seminarraum kann mehrere Beamer haben, von denen die Betriebsstunden gespeichert werden. Ein Beamer kann zu einem Zeitpunkt Teil von genau einem Seminarraum sein.

Lösung für I und II:

