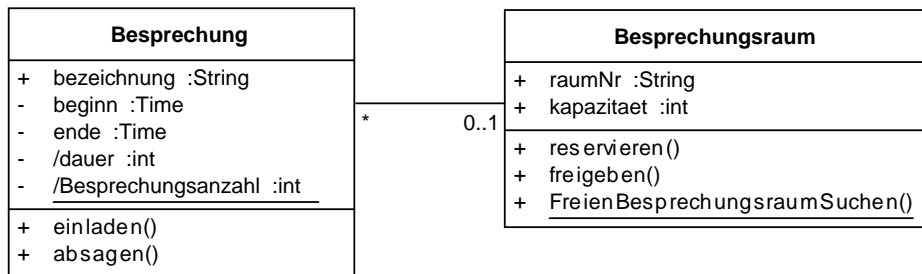


Hinweise: Die ersten vier Beispielen bestehen aus Modellierungsbeispielen und dazu passenden Theoriefragen.

- Modellierungsbeispiele: Bilden Sie den Sachverhalt, der in der Angabe geschildert wird, möglichst genau ab. Sollte etwas in der Angabe nicht erwähnt sein, treffen Sie sinnvolle Annahmen.
- Theoriefragen: Nehmen Sie sich bei der Beantwortung die Modellierungsaufgaben der jeweiligen Aufgabe zu Hilfe.

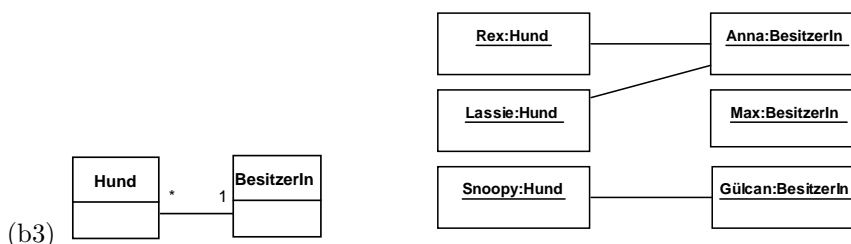
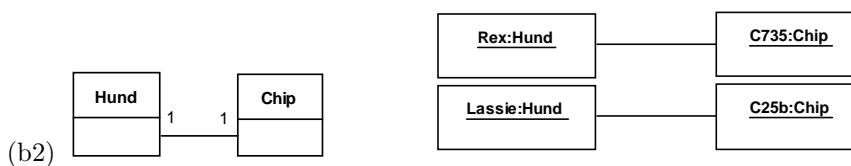
Aufgabe 1: Einleitende Beispiele, Teil 1

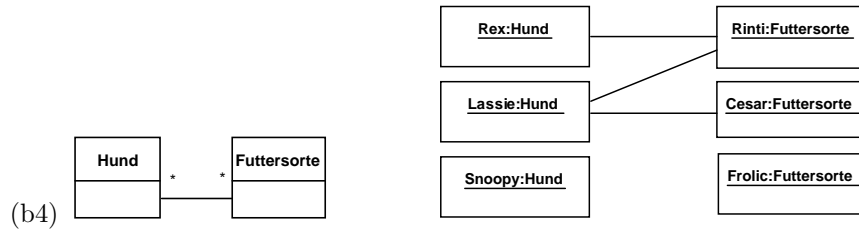
- a) *Modellierungsbeispiel:* Von jeder Besprechung werden die Bezeichnung, die Beginn- und Endezeit, die Dauer sowie die Anzahl an Besprechungen gespeichert. Man kann zu Besprechungen einladen und Besprechungen absagen. Eine Besprechung kann in einem Besprechungsraum stattfinden. Dieser hat eine Raumnummer und eine Kapazität. Man kann einen Besprechungsraum reservieren und diesen wieder freigeben. Weiters kann man einen freien Besprechungsraum suchen.



- b) *Modellierungsbeispiel:* Gegeben seien nachfolgende Ausschnitte aus Klassendiagrammen. Modellieren Sie zu jedem Ausschnitt ein passendes Objektdiagramm.

Zusatzfragen: Was ist ein Objekt? Erläutern Sie die Notation eines Objekts. Wo bzw. wie sieht man den Unterschied zwischen 1:1-, 1:n- und n:m-Assoziationen im Objektdiagramm?



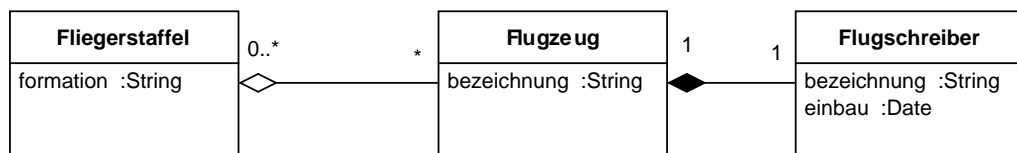


Aufgabe 2: Einleitende Beispiele, Teil 2

Modellieren Sie die Sachverhalte als Klassendiagramme und treffen Sie realistische Annahmen wo nötig:

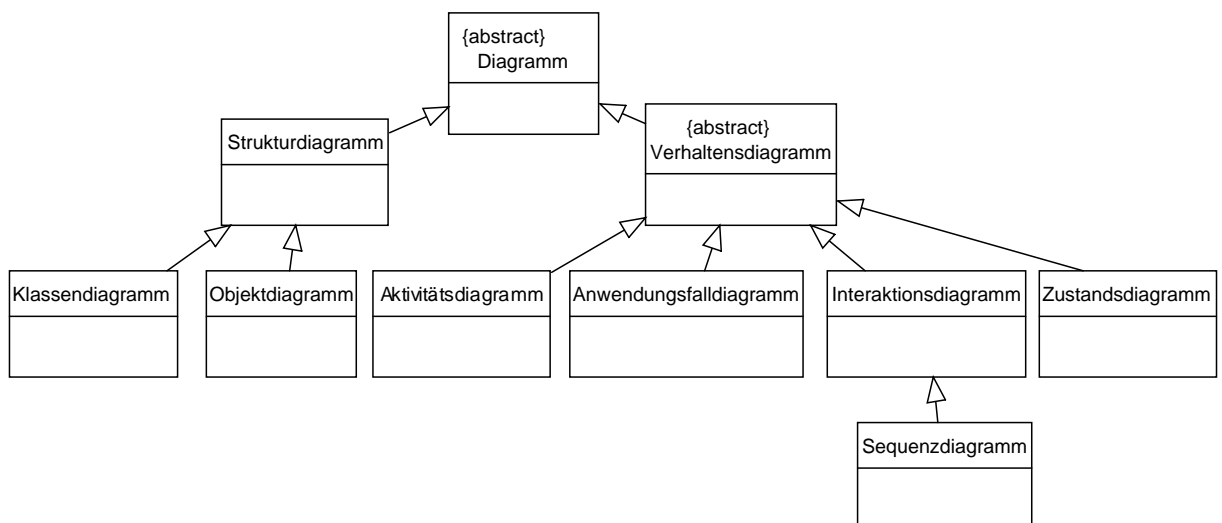
- a) *Modellierungsbeispiel:* Von jedem Flugzeug wird die Bezeichnung gespeichert. Ein Flugzeug kann Teil einer oder mehrerer Fliegerstaffeln sein; es wird der Name der Formation gespeichert, in der die Fliegerstaffel fliegt. In jedem Flugzeug ist ein Flugschreiber fix eingebaut und kann nicht mehr entfernt werden. Vom Flugschreiber wird die Bezeichnung und das Datum des Einbaus gespeichert.

Zusatzfragen: Beschreiben Sie den Unterschied zwischen starker und schwacher Aggregation. Erklären Sie die Notation.



- b) *Modellierungsbeispiel:* Es gibt genau zwei Arten von UML-Diagrammen, nämlich Strukturdiagramme und Verhaltensdiagramme. Klassendiagramm und Objektdiagramm zählen zu den Strukturdiagrammen. Es gibt vier verschiedene Verhaltensdiagramme: Zustandsdiagramm, Aktivitätsdiagramm, Anwendungsfalldiagramm und Interaktionsdiagramm. Das Sequenzdiagramm ist ein Beispiel für ein Interaktionsdiagramm.

Zusatzfragen: Was ist eine Generalisierung, was versteht man unter Mehrfachvererbung und was ist eine abstrakte Klasse? Erklären Sie die Notation.

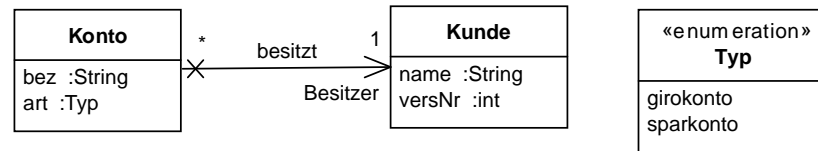


Aufgabe 3: Einleitende Beispiele, Teil 3

Modellieren Sie die Sachverhalte als Klassendiagramme und treffen Sie realistische Annahmen wo nötig:

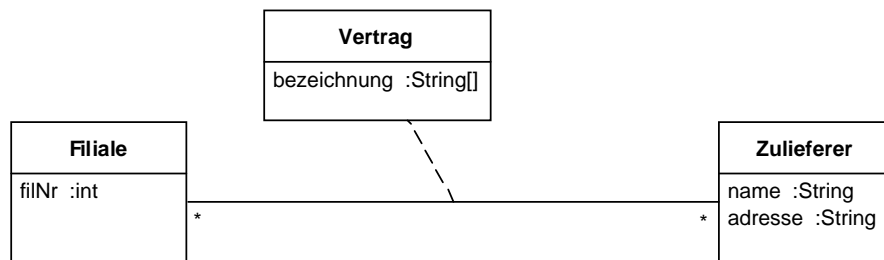
- a) *Modellierungsbeispiel:* Von jedem Konto wird die Bezeichnung und die Art des Kontos (girokonto oder sparkonto) gespeichert. Jeder Kunde wird mit Name und Sozialversicherungsnummer gespeichert und kann mehrere Konten haben, es lassen sich aber keine Rückschlüsse ziehen, welche Konten ein bestimmter Kunde hat. Der Kunde, der das Konto besitzt, wird als Besitzer bezeichnet.

Zusatzfragen: Erläutern Sie alle Notationsmöglichkeiten für eine Assoziation. Was versteht man unter einer Enumeration?



- b) *Modellierungsbeispiel:* Jede Supermarktfiliale hat mehrere Zulieferer die wiederum mehrere Filialen beliefern. Von jeder Filiale wird die Filialnummer gespeichert, vom Zulieferer sein Name und seine Adresse. Mit jedem Zulieferer, der eine Filiale beliefert, wird ein Liefervertrag abgeschlossen, in dem die Bezeichnungen der Produkte gespeichert werden.

Zusatzfragen: Was ist eine Assoziationsklasse? Wann ist es sinnvoll, eine solche einzusetzen? Erklären Sie die Notation.



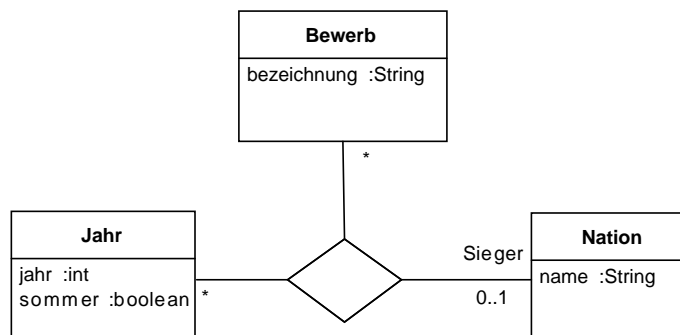
Modellieren Sie die Sachverhalte als Klassendiagramme und treffen Sie realistische Annahmen wo nötig:

- ```

classDiagram
 class Kindergarten
 class Mitarbeiter {
 Name :String
 }
 class Pädagogen
 class Teamleiterin {
 <<abstract>>
 }
 Kindergarten "*" -- "*" Mitarbeiter
 Pädagogen --|> Mitarbeiter
 Teamleiterin "*" -- "*" Mitarbeiter

```

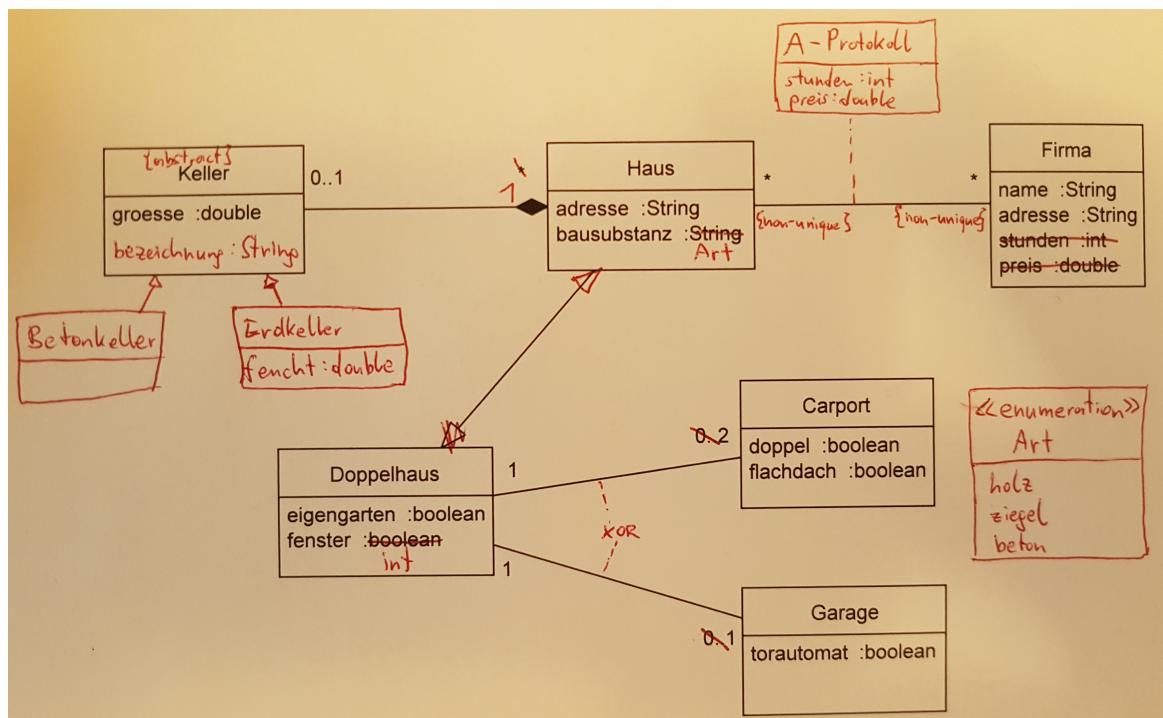
- Zusatzfragen:* Was ist eine n-äre Assoziation und wann wird sie benötigt? Erklären Sie die Notation.



### Aufgabe 5: Fehler finden

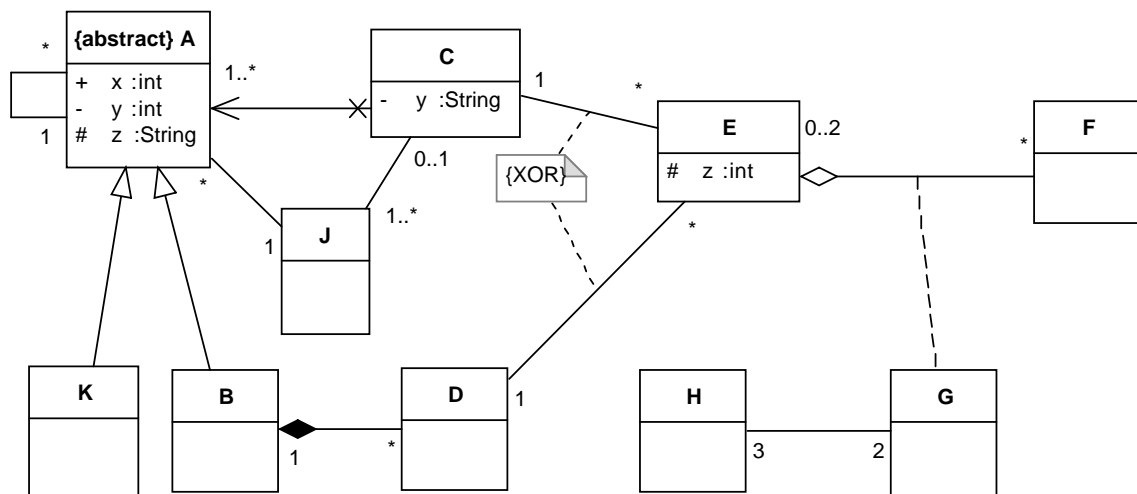
Gegeben ist folgendes UML Klassendiagramm. Bei der Modellierung sind leider einige Fehler passiert. Finden Sie die Fehler und korrigieren Sie diese im Diagramm.

Von jedem Haus werden Adresse und Bausubstanz (Holz, Ziegel oder Beton) gespeichert. Ein Haus kann einen Keller haben, von diesem werden die Größe sowie die Bezeichnung gespeichert. Es gibt genau zwei verschiedene Arten von Kellern, nämlich Erdkeller und Betonkeller. Von Erdkellern wird zusätzlich die Höhe der Luftfeuchtigkeit gespeichert. Ein spezielles Haus ist das Doppelhaus. Von ihm wird zusätzlich gespeichert, ob es einen eigenen Garten besitzt und wie viele Fenster es hat. Jedes Doppelhaus hat entweder zwei Carports oder eine Garage. Von Carports wird gespeichert ob es sich um Doppelcarports handelt und ob sie ein Flachdach haben, von der Garage wird gespeichert, ob sie ein automatisches Garagentor besitzt. Jedes Haus wird von mehreren Firmen betreut, von denen Name und Adresse gespeichert werden. Immer, wenn eine Firma in einem Haus einen Auftrag ausführt, wird die Anzahl der Stunden und der vereinbarte Preis in einem Auftragsprotokoll festgehalten.



## Aufgabe 6: Klassendiagramm lesen - Wahr oder falsch?

Es ist folgendes UML-Modell gegeben:



Welche Aussagen treffen zu? Begründen Sie Ihre Antwort!

|                                                                                               |                                          |
|-----------------------------------------------------------------------------------------------|------------------------------------------|
| Eine Instanz von C kann auf die Variable <i>x</i> zugreifen.                                  | <input checked="" type="checkbox"/> ja   |
| Die Raute bei B wird als Komposition bezeichnet.                                              | <input checked="" type="checkbox"/> ja   |
| Wenn eine Instanz von E gelöscht wird, werden alle enthaltenen Instanzen von F gelöscht.      | <input checked="" type="checkbox"/> nein |
| Ein Objekt von E steht in Beziehung zu genau einem Objekt von C und genau einem Objekt von D. | <input checked="" type="checkbox"/> nein |
| Ein Objekt von C steht in Beziehung zu mindestens einer direkten Instanz von A.               | <input checked="" type="checkbox"/> nein |

|                                                                       |                                          |
|-----------------------------------------------------------------------|------------------------------------------|
| Ein Objekt von B kann mit sich selbst in Beziehung stehen.            | <input checked="" type="checkbox"/> ja   |
| Ein Objekt von A kann auf die Variable <i>z</i> zugreifen.            | <input checked="" type="checkbox"/> ja   |
| Ein Objekt von B muss in Beziehung zu einem Objekt von J stehen.      | <input checked="" type="checkbox"/> ja   |
| Ein Objekt von J steht in Beziehung zu mindestens einem Objekt von A. | <input checked="" type="checkbox"/> nein |
| Im System können mehr G als F enthalten sein.                         | <input checked="" type="checkbox"/> ja   |

|                                                                                                                         |                                          |
|-------------------------------------------------------------------------------------------------------------------------|------------------------------------------|
| Ein Objekt von E muss in Beziehung zu einem Objekt von C stehen.                                                        | <input checked="" type="checkbox"/> nein |
| Ein Objekt von B kann direkt auf die Variable <i>z</i> zugreifen.                                                       | <input checked="" type="checkbox"/> ja   |
| Ein Objekt von D ist in genau einem Objekt von B enthalten.                                                             | <input checked="" type="checkbox"/> ja   |
| Zwei Objekte von G stehen mit drei Objekten von H in Beziehung.                                                         | <input checked="" type="checkbox"/> nein |
| Ein Objekt von C steht in Beziehung zu mindestens einem Objekt von A und die Beziehung kann von C aus navigiert werden. | <input checked="" type="checkbox"/> ja   |