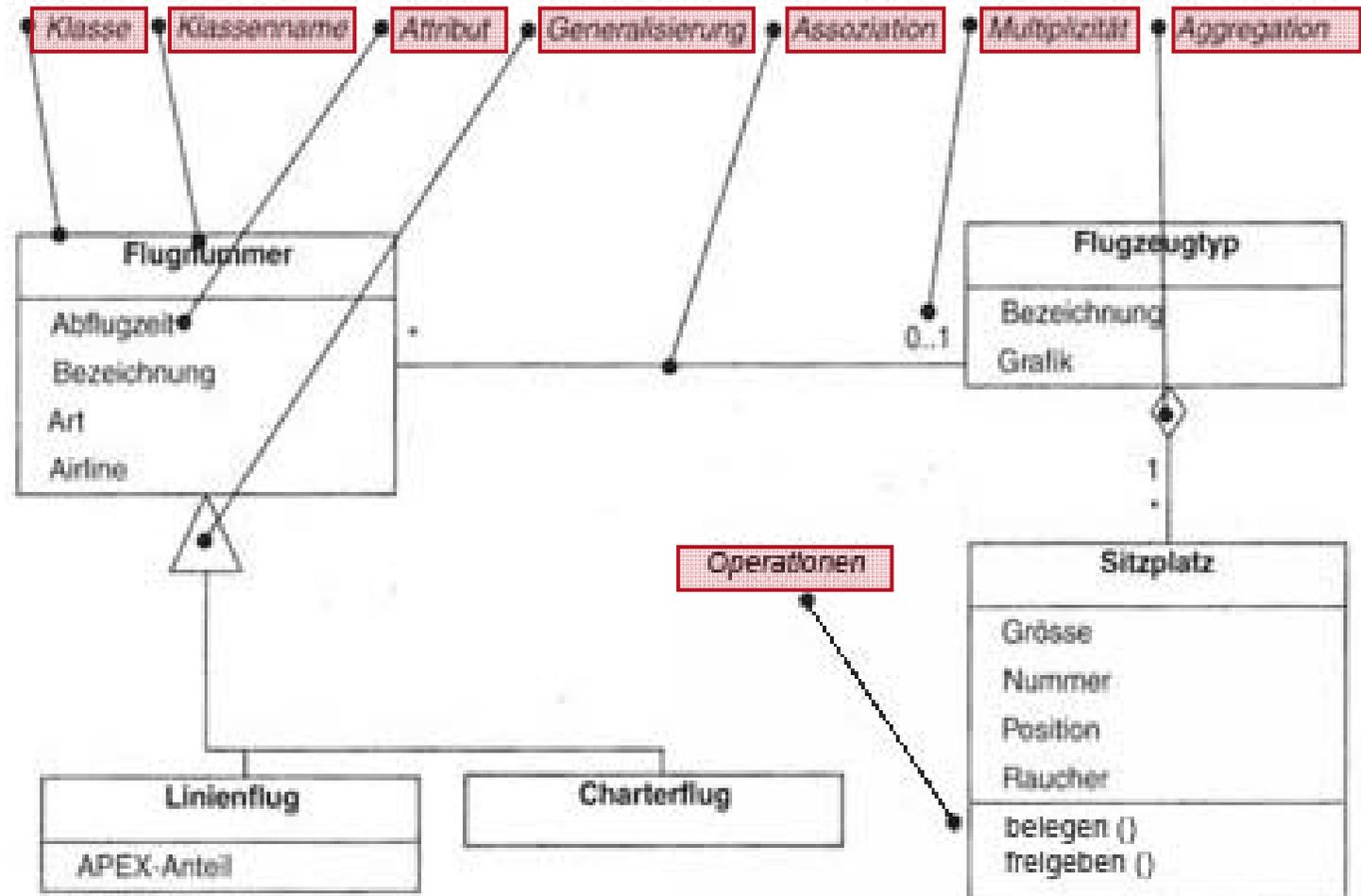


## Klassendiagramm




**BlaBla**

Diese Kennzeichnungen sind nur Erläuterungen und nicht Bestandteil des Diagramms

Quelle: P.Grässle, H.Baumann, P.Baumann, UML projektorientiert, Galileo Verlag, 2003

### Klassendiagramm

#### Primäre Begriffe

- Kapselung
  - Klassen fassen Attribute und Operationen zusammen.
  
- Objekte aus Klassen:
  - Klassen (Class) trennen die generelle Beschreibung der Struktur und des Verhaltens von den tatsächlich instantiierten Objekten
  
- Objekt-Identität:
  - Auch von der gleichen Klasse instantiierte und die gleichen Attributwerte enthaltenden Objekte sind unterscheidbare individuelle Objekte.
  
- Kohärenz:
  - Jede Klasse sollte möglichst nur genau einen sachlogischen Aspekt widerspiegeln und dafür verantwortlich sein.
  - Eigenschaften sollten nicht wegen Optimierungen und Redundanzfreiheit einer Klasse zugeordnet werden, wenn sachlogisch/fachlich eine andere Klasse zuständig ist.

### Klassendiagramm

#### Primäre Begriffe

- Vererbung, Spezialisierung:
  - Klassen können hierarchisch strukturiert werden und Eigenschaften an Unterklassen vererben (Inheritance).
  - Entsprechend einer Spezialisierung erben Unterklassen alle Eigenschaften der Oberklasse und fügen weitere Eigenschaften hinzu.
  - Unterklassen können die geerbten Eigenschaften der Oberklasse überschreiben und erweitern, jedoch nicht eliminieren oder unterdrücken.
  
- Substitution:
  - Objekte von Unterklassen können jederzeit anstelle von Objekten ihrer Oberklasse eingesetzt werden.
  
- Statische Polymorphie, Überladen:
  - Es können mehrere gleichnamige Operationen (Methoden) innerhalb einer Klasse definiert werden, wenn sich deren Signatur unterscheidet, also wenn die verwendeten Methodenparameter von unterschiedlichen Typen sind.

### Klassendiagramm

#### Primäre Begriffe

- Dynamische Polymorphie, Überschreiben:
  - Wird eine Methode in mehreren von der gleichen Vorfahrenklasse abgeleiteten Klassen überschrieben und unterschiedlich implementiert und auf diese Methoden über eine Variable vom Typ der Vorfahrenklasse zugegriffen, nennt man das dynamische Polymorphie.
  
- Persistenz, Transienz:
  - Persistente Objekte sind serialisierbar und werden langfristig gespeichert (z.B. im Dateisystem oder in einer Datenbank).
  
  - Transiente Objekte werden nur vorübergehend benötigt und nicht gespeichert und gehen bei Programmende verloren

### Klassendiagramm – Primäre Notationen

- Klassennotation

Drei Rubriken:

1. Klassenname (eventuell mit Zusätzen), z.B. Objektname:Klassenname
2. Attribute (Eigenschaften / Daten), z.B. Sichtbarkeit Attribut:Typ=Wert
3. Operationen (Methoden), z.B. Sichtbarkeit Methode(Parameter):Returntyp)

Vor den Attribut- oder Methodennamen können Sichtbarkeitssymbole stehen: + für public, - für private und # für protected. Klassenattribute und -methoden werden unterstrichen. Beispiele für Klassen sind Passagier, Flugzeug, Ticket.

Die Notation für Objekte ist ähnlich der Klassennotation. Statt des Klassennamens wird der unterstrichene Objektname eingesetzt, eventuell gefolgt von einem Doppelpunkt und dem Namen der instantiierten Klasse ("objektName: Klassenname")

- Die Assoziation stellt eine allgemeine Beziehung zwischen zwei Klassen dar - über die Realisierung wird dabei nichts ausgesagt.
- Eine besondere Assoziation ist die Aggregation, die durch eine Raute an der Linie dargestellt wird (z. B. zwischen Flugzeugtyp und Sitzplatz). Sie gibt an, dass eine Klasse Sitzplatz in der Klasse Flugzeugtyp "enthalten" ist (Ist-Teil-von-Beziehung).

### Klassendiagramm – Primäre Notationen

- An einer Assoziation können noch Multiplizitäten, d. h. Zahlen oder Zahlbereiche, angegeben werden. Diese bestimmen die Anzahl der Objekte, die miteinander in Beziehung stehen. Beispielsweise ist in einem Flugzeugtyp 1 bis n Sitzplätze vorhanden.
- Die Komposition ist eine stärkere Form der Aggregation, die durch eine ausgefüllte Raute dargestellt wird. Bei der Beziehung Komposition handelt es sich um ein physikalisches Enthaltensein.
- Vererbung (Inheritance)  
Der Pfeil ist geschlossen und durchgezogen und zeigt von der abgeleiteten Unterklasse (= Subklasse) zur Oberklasse (= Basisklasse = Superklasse). Die Oberklasse ist eine Generalisierung der Unterklasse und umgekehrt ist die Unterklasse eine Spezialisierung der Oberklasse ("Oberbegriff"-Beziehung = "Ist-ein"-Beziehung).

### Vorgehensweise zur Erstellung der UML-Diagramme

#### Beispiel-Ablauf

- Idee entsteht
  - Use Cases identifizieren und spezifizieren
  - Aktivitätsdiagramme anfertigen
  - Klassen identifizieren
  - Assoziationen in Klassendiagramm identifizieren
    - Nur als Linie eingetragen
  - Attribute in Klassendiagramm identifizieren
    - Attribute eintragen
  - Vererbung in Klassendiagramm identifizieren
    - Vererbungsstrukturen eintragen
  - Assoziationen in Klassendiagramm spezifizieren
    - Kardinalitäten, Aggregationen, Kompositionen definieren
  - Attribute in Klassendiagramm spezifizieren
  - Sequenzdiagramme
    - Für jeden Weg im Use Case ein Sequenz-Diagramm
  - Zustandsdiagramme erstellen
    - Für jeden nicht-trivialen Lebenszyklus einer Klasse
  - Operationen in Klassendiagramm spezifizieren
- UML-Modell entsteht