

Dies ist keine „Musterlösung“, sondern eine gute von vielen möglichen Lösungen. Kommentare, die nicht Teil der Lösung sind, sind *kursiv* gesetzt.

**Aufgabe 2-1:**

1. Es gibt natürlich viele, auch recht verschiedene Definitionen. Diese hier sind zu den Übungsblättern und der Vorlesung konsistent, nicht zwingend zu ALP.

Klasse: Bezeichnung für Sprachkonstrukte, mit denen verschiedenartige Elemente (members) mit gemeinsamem Gültigkeitsbereich zusammengefasst werden können. Elemente können Methoden oder Attribute sein. Klassen haben in ihren Attributen Zustände, im Gegensatz zu abstrakten Datentypen, die zwar Operationen kennen, aber keinen Zustand haben (d.h. insbesondere keine Daten speichern können).

Vererbung: Eine Klasse kann Elemente an eine Unterklasse vererben. Die Unterklasse ist dann eine Erweiterung oder Spezialisierung der Oberklasse, die Oberklasse eine Generalisierung der Unterklasse.

Attribut: Die Attribute einer Klasse beschreiben Zustände der Objekte dieser Klasse.

Methode = Operation: Eine Methode/Operation einer Klasse beschreibt ein Berechnungsverfahren unter zu Hilfenahme der Attribute einer Klasse.

Exemplar = Objekt: Ein Objekt eine konkrete Ausprägung einer Klasse zur Laufzeit des Programms. Ein weiterer Begriff für dasselbe ist Ausprägung; das schöne an diesem ist, dass es auch ein Verb dazu gibt. Man sagt nicht „Instanz“ (bedeutet im Deutschen was andere) oder „instanzieren“ (das Wort gibt es im Deutschen nicht) dazu!

Bibliothek: Eine Bibliothek fasst eine Menge Klassen und Methoden zwecks Wiederverwendung zusammen.

Spezifikation einer Methode = Beschreibung ihres Kopfes („Syntax“) und ihres von außen sichtbaren Verhaltens („Semantik“) unter vollständiger Abstraktion von ihrer Implementierung (Rumpf). Eine Spezifikation beantwortet die Fragen „WAS leistet das System?“ oder „Was muss der Benutzer tun und was kann er erwarten?“ In der Softwaretechnik wird deutlich mehr spezifiziert als nur Methoden!

Implementierung beantwortet die Fragen „WIE erbringt das System seine Leistung?“ „Wie ist das System konstruiert?“

Verifikation = Nachweis, dass die Implementierung der Spezifikation entspricht.

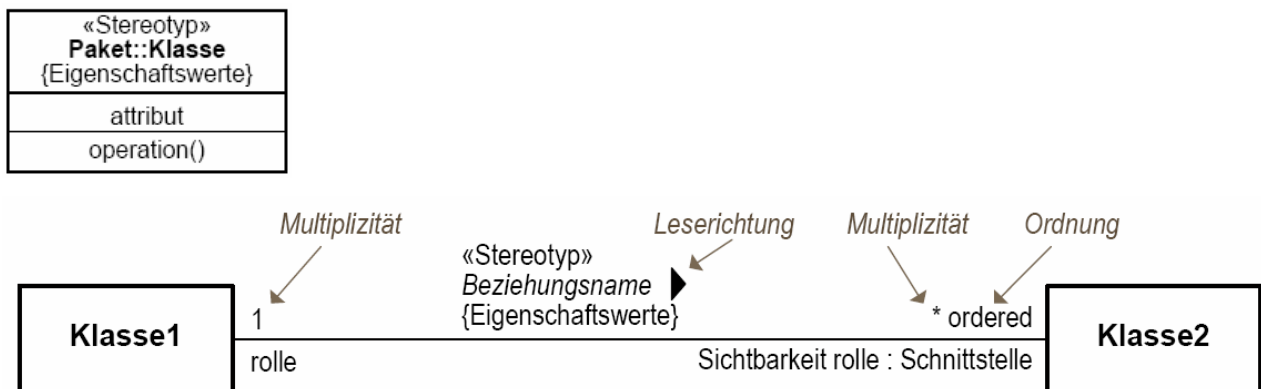
Geheimnisprinzip (information hiding principle): Lokale Entwurfsentscheidungen innerhalb eines Moduls (z.B. einer Klasse) sollten von außen nicht sichtbar sein. Kenntnis einer Spezifikation des Moduls ist notwendige und hinreichende Voraussetzung für seine korrekte Benutzung. Das Geheimnisprinzip wird in der Vorlesung noch wichtig!

2. Vor allem Anforderungserhebung und (Projekt-)Management, sie waren bislang kein explizites Thema im Studium. Die Aktivitäten stehen auf Folie 21.

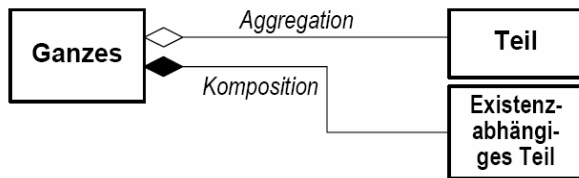
**Aufgabe 2-2:**

Eine ähnliche Lese- und Verstehaufgabe eines Klassendiagramms gab es mal in einer SWT-Klausur.

1.a. Siehe Grafiken. Multiplizität wird in anderen Gebieten Kardinalität genannt. Bitte auf die Leserichtung der Multiplizitäten achten, das wird immer wieder falsch gemacht! Beispiel: Wenn ein Kind zwei Eltern hat, steht die 2 bei der Klasse „Elter“. Hinweis zu den Beziehungen: Es können bei einer Beziehung zwischen Klassen (eine Unterklassenbeziehung nenne ich nicht so - sie heißt in UML Vererbung) drei Namen stehen: Je ein Name an der Klasse für deren Rolle, die sie in dieser Beziehung spielt (Person kann Patient für den Arzt sein und Versicherter für die Krankenkasse sein) und ein Name für die Beziehung selbst - die in der Mitte der Kante steht. Gemeint war der Name der Beziehung selbst. Auf dem eGK-Diagramm taucht ein solcher Name leider nirgends auf.



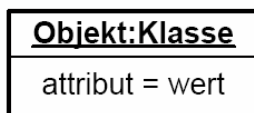
1.b. Es zeigt eine Teil-von-Beziehung an. Ein Objekt a der Klasse Ganzes hat Objekte b der Klasse Teil (Anzahl entspr. der Multiplizität) als Teil (d.h. die b „sterben“, wenn auch a „stirbt“), wenn an Ganzes die schwarze Raute notiert ist als Abschluss der Beziehungskante zwischen Ganzes und Teil.



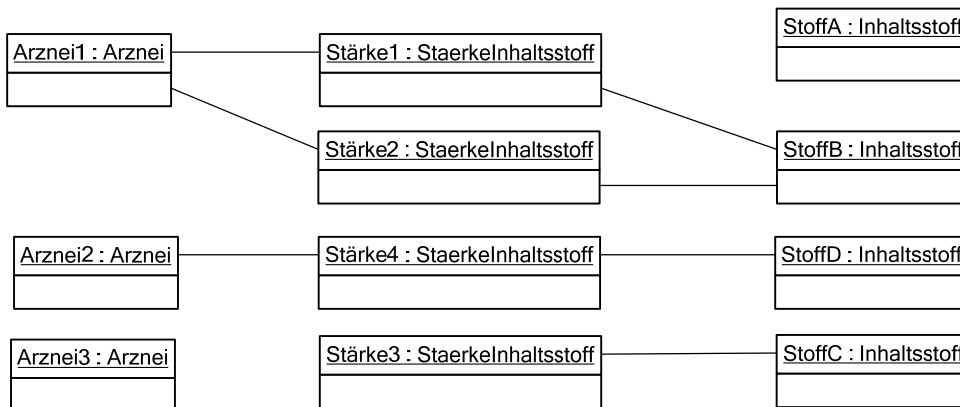
Im Gegensatz zu den einfachen Beziehungen („Assoziation“) nennt sich dies „Komposition“. Es gibt noch die Aggregation mit leerer Raute  $\diamond$  - die kommt in einem späteren Übungsblatt. Beispiel: Das Gehirn ist Teil (Komposition) eines Menschen (wenn der Mensch stirbt, stirbt auch das Gehirn), die Brille ist nur assoziiert mit dem Menschen.

Die Bilder sind von <http://www.oose.de/downloads/uml-notationsuebersicht.pdf>

2. Ein Rezept ist eine geordnete, nicht-leere Liste von Rezeptpositionen. Jede Rezeptposition ist Teil genau eines Rezepts und kann eine Dosierung als Einnahmeangabe haben.
- 3.a. Ein Objektdiagramm stellt Ausprägungen von Klassen und deren Ausprägungen von Beziehungen dar. Objekte sind Rechtecke, der Name ist individuell (mit Angabe der Klasse hinter:) und unterstrichen.



- 3.b. Dies zeigt die essentiellen Möglichkeiten.



3.c. Dass ein Inhaltsstoff in zwei verschiedenen Mengen in einer Arznei sein kann (siehe im Beispiel StoffA in Arznei1) ist sonderbar. Gibt es aber vielleicht Zwei-Phasen-Medikamente (Anti-Baby-Pille?), die den gleichen Wirkstoff in zwei Dosierungen hintereinander abgeben? Ebenfalls sonderbar ist, dass es Arzneien ohne Inhaltsstoff geben kann (siehe im Beispiel Arznei3). (Vielleicht ja Placebos? Werden die wirklich verschrieben?) Aber da uns das Domänenwissen fehlt (wir sind keine Pharmazeuten), kann man sich nicht ganz sicher sein, ob es doch Sinn macht. Es ist auch nicht grundsätzlich schlimm, wenn eine UML-Spezifikation etwas in der Realität nie vorkommendes zulässt. Häufig sind solche Diagramme durchaus unterspezifiziert, denn oftmals ist die Lesbarkeit auf dieser Abstraktionsebene wichtiger zu werten. Wir werden später in der Vorlesung OCL kennen lernen, mit der man genauer spezifizieren kann.

- 4.a. Ja. (Begründung anhand des Diagramms fehlt.)

- 4.b. Nein, es gibt zwei Fehler:

1. „Zwei Einnahmeangaben“ geht nicht, es kann maximal eine geben.
2. Wegen Komposition und Multiplizität kann ein Dosierungsschema nur zu genau einer Dosierung gehören und nicht zu zweien.

Achtung: Es scheint so, als hätten Einzelpositionen gar keine Dosierung, sondern nur Rezeptpositionen. (Wer auf diesen Gedanken kommt, ist schon mal zu loben!) Aber da diese Einzelpositionen sind, \*gibt\* es also Einzelpositionen, die Dosierungen haben - wenn auch wohl nicht alle. Aber: Wegen des kursiven Namens ist Einzelposition \*abstrakt\*, d.h. Einzelposition kann nicht ausgeprägt werden bzw. es gibt keine Objekte, die Einzelpositionen sind, aber nicht eine ihrer Unterklassen. Da es keine anderen Unterklassen gibt, sind also im Rahmen dieses Diagramms (closed world assumption) alle Einzelpositionen Rezeptpositionen und haben somit eine Dosierung.

### Aufgabe 2-3:

Jeder hat eigene Vorstellungen. In den meisten InformatikerInnenjobs geht es aber in irgendeiner Weise um die Erstellung von größerer Software und damit automatisch um Softwaretechnik. Es kann aber sein, dass grad der erste Job sehr kodiernah ist und somit ALP relevanter ist. Das ändert sich mit der Karriere und dem zunehmenden Offshoring (= Verlagerung von Jobs in Billiglohnländer) aber ganz bestimmt!