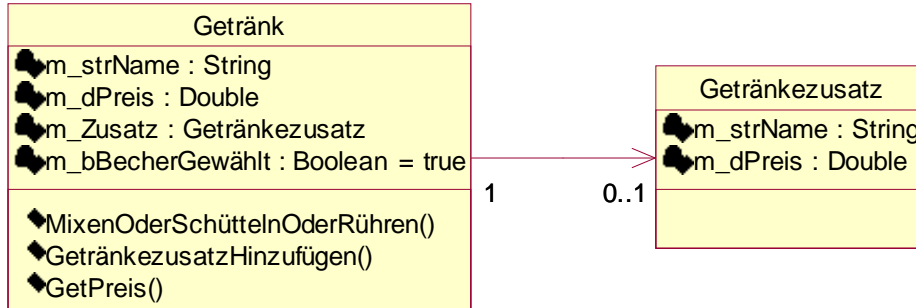


**Aufgabe 1**

- a) Ich halte es für sinnvoll, die Getränkezusätze als eigenständige Objekte zu modellieren, was man in UML so ausdrücken könnte:

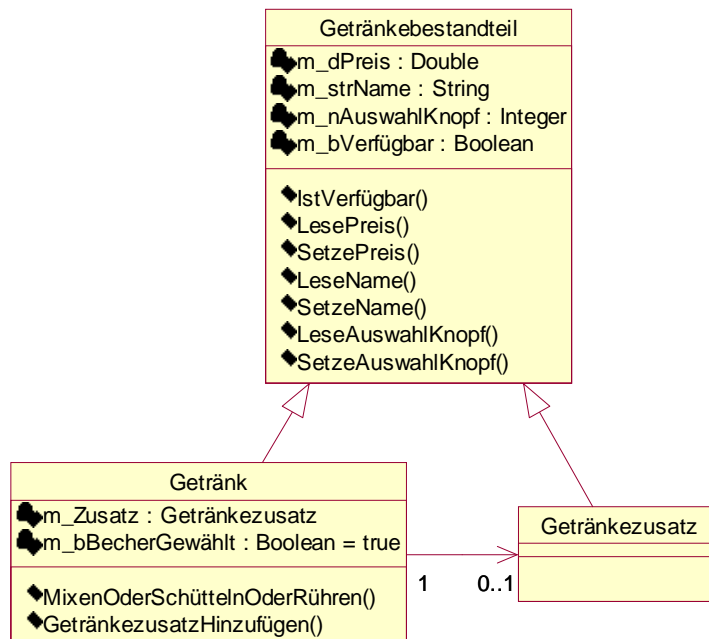


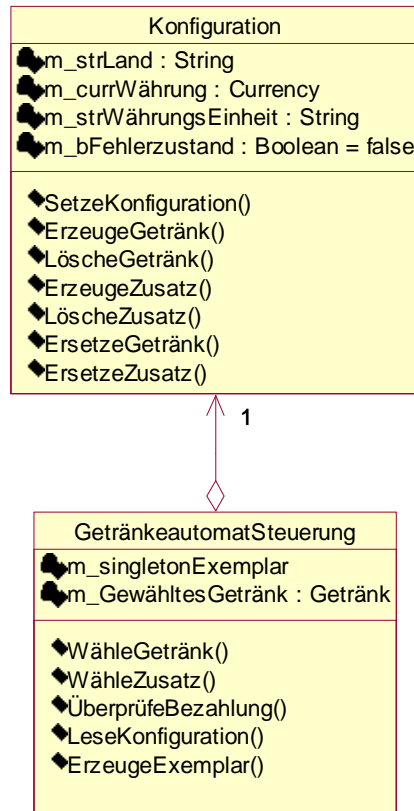
Der wesentliche Vorteil besteht darin, dass man flexibel in Hinblick auf zukünftige Änderungen ist. Es ist vorstellbar, dass z.B. ein vollkommen neuer Zusatz entwickelt wird. Hat man Getränkzusatz lediglich als Attribut implementiert, so kann dies nicht mehr berücksichtigt werden, da man den Wertebereich von Zusatz im kompletten Code ändern müsste. Hingegen kann die Lösung auf Basis einer eigenen Getränkezusatz-Klasse beliebig manipuliert werden, da man nur zu garantieren braucht, dass die minimale und abstrakte Schnittstelle implementiert ist.

Sollte man genau wissen, dass nie ein neuer Getränkezusatz hinzugefügt wird, so ist es effizienter, doch auf ein einfaches Attribut zurückzugreifen.

- b) *Hinweis:*

Die nun folgenden UML-Diagramme sind ursprünglich von Herrn Gericke entworfen worden. Ich habe lediglich einige Umbenennungen vorgenommen bzw. die graphische Anordnung geändert. Wie schon in Aufgabe 1 orientiere ich mich auch dieses Mal an der Ungarischen Notation für Variablenbezeichner, wie sie auch Microsoft verwendet.

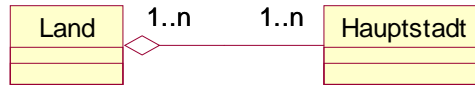




**Aufgabe 2**

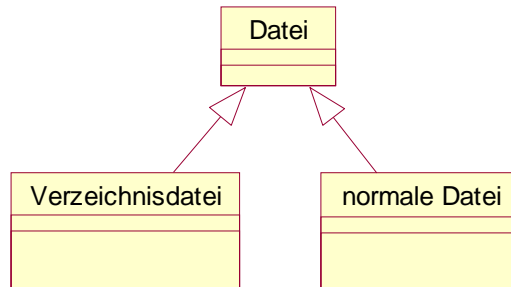
*Ein Land hat mindestens eine Hauptstadt.*

Eine Aggregation ist angebracht, da die Hauptstadt in Hinblick auf ihr Land einer ist-Teil-von-Beziehung unterliegt.



*Eine Datei ist eine normale Datei oder eine Verzeichnisdatei.*

Sowohl eine normale Datei als auch eine Verzeichnisdatei sind eine Spezialisierung einer Datei. Dadurch bietet sich eigentlich nur eine Vererbungsbeziehung an.



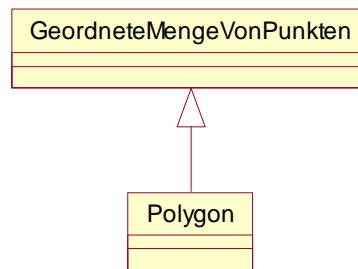
*Eine Datenbank enthält Datensätze.*

Die Datenbank selbst ist lediglich eine Verwaltungsstruktur, die für die Organisation der Datensätze zuständig ist. Es liegt weder eine Vererbungsbeziehung noch eine Aggregation vor, hier reicht eine einfache Assoziation aus.



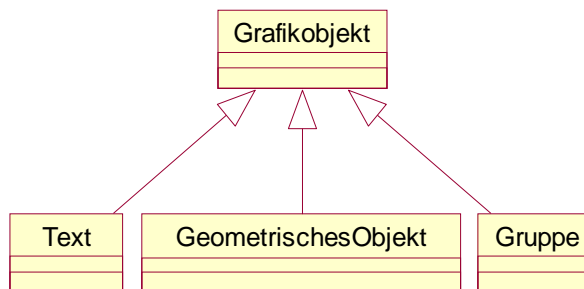
*Ein Polygon ist durch eine Menge von geordneten Punkten definiert.*

Es liegt wieder eine Spezialisierung vor, d.h. eine Vererbung ist zu verwenden.



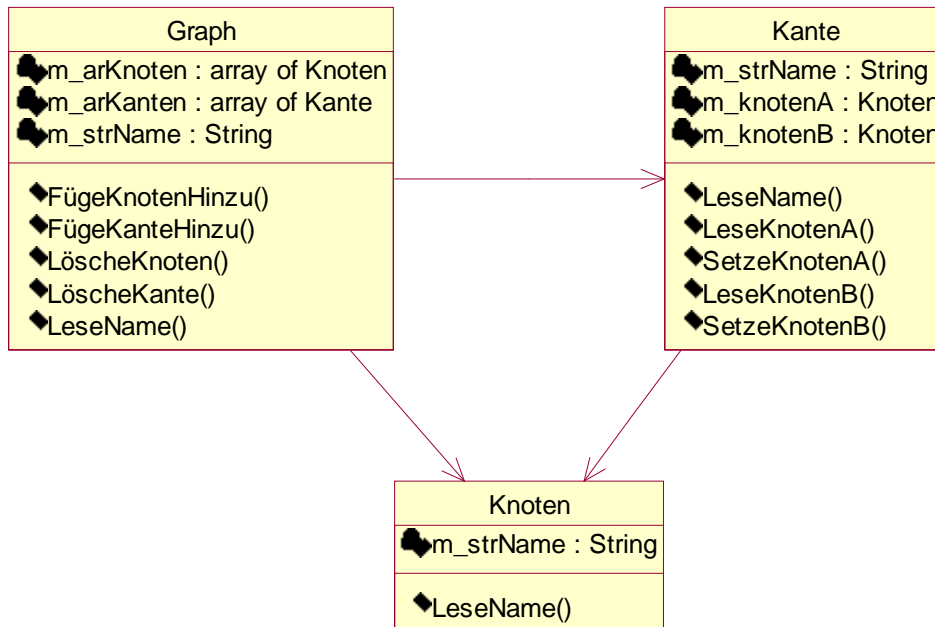
*Ein Grafikobjekt ist ein Text, ein geometrisches Objekt oder eine Gruppe.*

Wiederum wird Grafikobjekt spezialisiert, was vernünftigerweise nur mit einer Vererbung zu lösen ist.



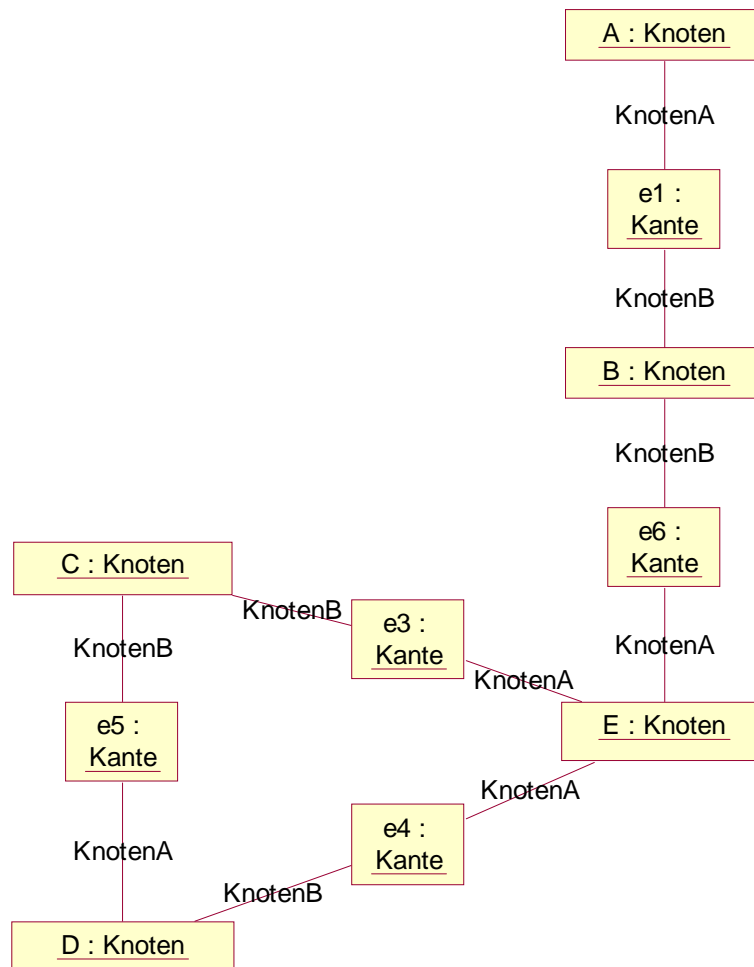
**Aufgabe 3**

Ein ungerichteter Graph lässt sich durch das gleiche Klassendiagramm wie ein gerichteter Graph darstellen, man muss lediglich die genaue Semantik der Bezeichnungen `KnotenA` und `KnotenB` einer Kante klären.



In einem ungerichteten Graph sind A und B beliebig austauschbar, in einem gerichteten nicht. Dies wirkt sich u.a. auf den Vergleich zweier Kanten aus, der dann unterschiedlich implementiert werden muss. Im Klassendiagramm ist es aber nicht notwendig, dass solche Details explizit geklärt werden.

Umgesetzt auf die gegebene Abbildung aus der Aufgabenstellung entsteht dann folgendes Objektdiagramm (leider ist das benutzte Modellierungstools in dieser Hinsicht sehr beschränkt):



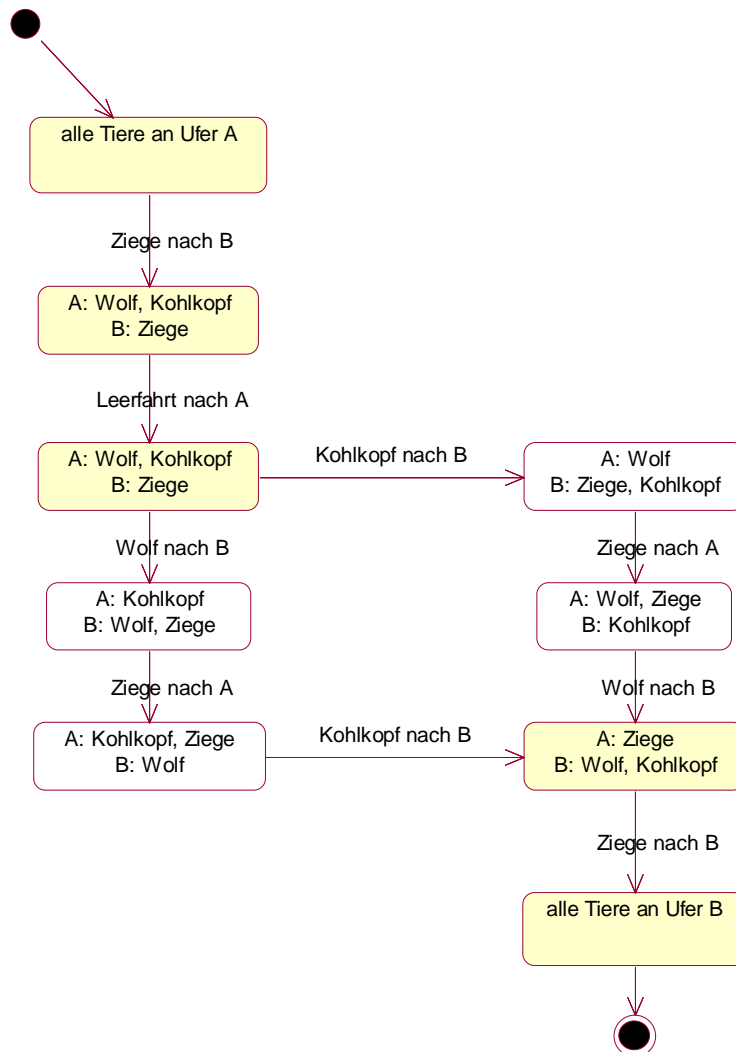
**Aufgabe 4**

Ich erachte es als sinnvoll, dass man – bevor man den Automaten konstruiert – die möglichen Kombinationen ermittelt, in denen niemand geärgert, gequält oder gefressen wird. Leider ist dies für die ersten beiden Verben nicht exakt verifizierbar, nur dem Genozid (rot markiert) kann Einhalt geboten werden.

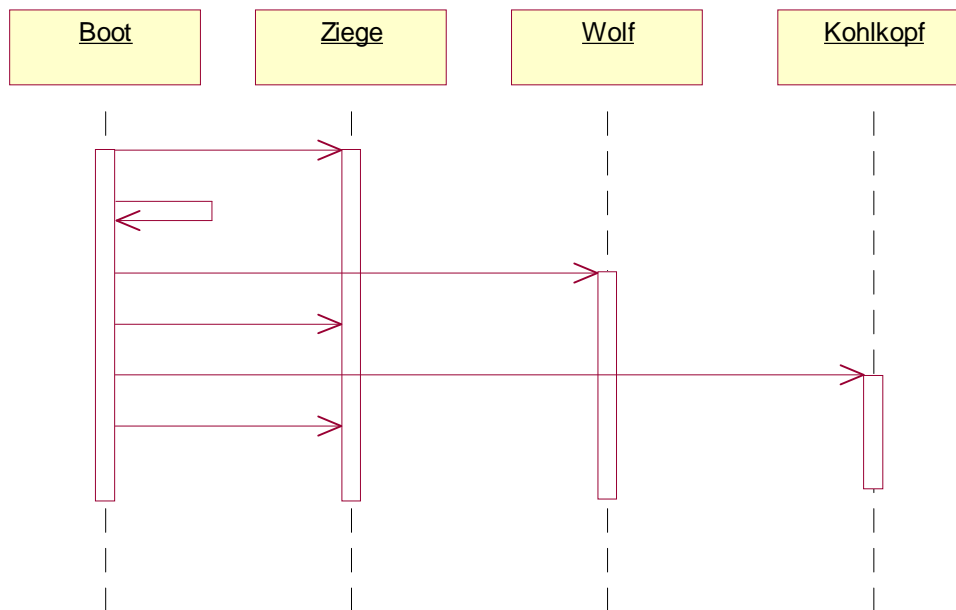
	Kohlkopf	Ziege	Wolf
Kohlkopf			
Ziege			
Wolf			

Man stellt sehr schnell fest, dass nur der Kohlkopf mit der Wolf harmoniert, die Ziege sollte von beiden fern gehalten werden. Wenn man sich den modellierten Automaten nun genauer ansieht, wird man feststellen, dass die weißen Zustände jedoch genau die eben erkannten tödlichen Situationen beinhalten. Sie sind keine „Verweilzustände“ sondern werden in unendlicher kurzer Zeit passiert, so dass kein zu verschiffendes Objekt Hunger bekommt.

Es existieren genau zwei Möglichkeiten, um Wolf, Ziege und Kohlkopf über den Fluss zu bringen.



Um den Umgang mit dem Modellierungstool zu verbessern, habe ich noch die Transportmöglichkeit A (linker Strang im vorherigen Automaten) als Sequenzdiagramm dargestellt:



Durch die explizite Hinzunahme des Objektes `Boot` kann man die kritischen Situationen im Automaten umgehen, da man sich auf den Transport als wesentlichen Vorgang konzentriert. Man sieht schnell, dass das Boot 6 Mal fahren muss und dabei die Ziege 3 Mal transportiert, u.a. ist auch eine Leerfahrt möglich. Im Automaten hätte man das erst nach genauerem Hinsehen erkannt, dafür ist dort die Verzweigung (aufgrund der beiden möglichen Varianten nach Zustand 3) einfacherer darzustellen.

*Der zweite Teil der Aufgabe wurde nicht bearbeitet.*

### Aufgabe 5

*(fehlt)*