



# Theorie: Modellierung/Diagrammtypen

## Was ist Modellierung? Wozu benutzt man Modellierung?

Modellierung ist das Abbilden eines Ausschnitts der Realität in einem Modell. Je nach Problemstellung werden unterschiedliche Teile der Realität in Modellen abgebildet. In der Informatik gibt es dazu verschiedene Modelle, die benutzt werden, um Informationen kompakt und übersichtlich darzustellen. Jedes Modell erfüllt einen bestimmten Zweck: Beschreibung von Zuständen und Eigenschaften von Objekten, Darstellung von Beziehungen zwischen Objekten, Erläutern von Abläufen und Prozessen usw. Die informatischen Modelle können Informationen aus vielfältigen Kontexten darstellen. Die Gegenstände dieser Modelle sind reale oder fiktive Objekte innerhalb oder außerhalb der Informatik. Daher können die Modelle auch in Unterrichtsfächern abseits des Fachs Informatik verwendet werden. Sie können z.B. für folgende Aufgaben in den einzelnen Unterrichtsfächern benutzt werden:

- Beschreibung von Berechnungsalgorithmen im Fach Mathematik oder Abläufen in naturwissenschaftlichen Fächern
- Ordnen und Beschreiben von geometrischen Objekten in den Fächern Mathematik und Geometrisches Zeichnen
- Zusammenfassung der wesentlichen Aspekte eines Textes im Fach Deutsch oder in einer Fremdsprache
- Üben von Vokabeln und Erweitern des Wortschatzes in einer Fremdsprache
- Planung von Abläufen oder Beschreibung von Prozessen in naturwissenschaftlichen Fächern.

Ein entscheidender Aspekt für den Unterricht sind die vielfältigen Visualisierungsmöglichkeiten von Informationen und die Komprimierung der Information auf die relevanten Inhalte.

Dieser Beitrag gibt einen Überblick über verschiedene Diagrammtypen/Modelle, die in der Informatik verwendet werden, und die Informationen, die durch diese Modelle darstellbar sind. Außerdem werden exemplarische Beispiele präsentiert.

Ein weit verbreiteter Standard für die grafische Darstellung von Modellen in der Informatik ist UML (Unified Modeling Language). In UML gibt es verschiedene Modelle mit festgelegter Notation, welche für eine Vielzahl von Anwendungsfällen geeignet sind. Zu den Diagrammarten von UML gehören u.a. die Objekt-, Klassen- und Aktivitätsdiagramme. Neben diesen UML-Diagrammen werden aber auch andere grafische Darstellungen von Modellen benutzt, etwa das Gegenstand-Beziehung-Modell (auch Entity-Relationship-Diagramm genannt).

## Modellarten

Je nachdem, welche Informationen der Realität man abbilden möchte, ist ein anderes Diagramm geeignet.

Grundlegend lässt sich zwischen Diagrammen unterscheiden, die einen Ablauf festlegen (wie z.B. das Aktivitätsdiagramm) und solchen, die z.B. statische Eigenschaften und Beziehungen von Gegenständen oder Personen beschreiben. Bei der zweiten Gruppe lässt sich zusätzlich unterscheiden, ob man ein konkretes Objekt (z.B. *mein Pferd*) oder eine Objektklasse (z.B. *das Pferd*) beschreiben möchte.

Je nachdem, welche Informationen man darstellen möchte, kann eines (oder auch mehrere) der unten angeführten Diagramme geeignet sein.

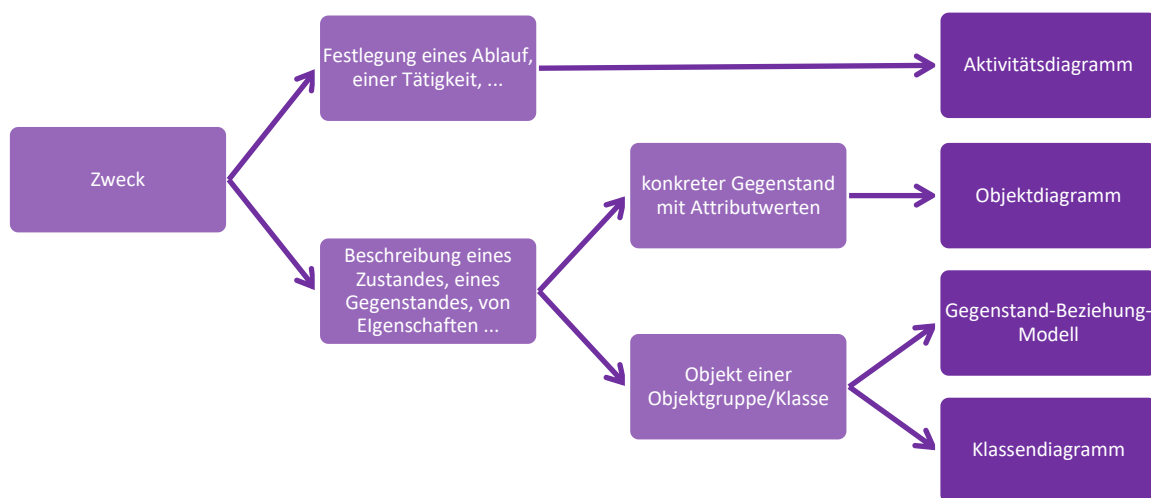


Abbildung 1: Diagrammtypen

- **Objektdiagramm:** Das Objektdiagramm stellt konkrete Objekte (z.B. *mein Pferd*) mit ihren Eigenschaften (z.B. *Alter*) und den Ausprägungen der Eigenschaften (z.B. *2 Jahre*) dar. Weiters können Beziehungen zwischen Objekten dargestellt werden.
- **Gegenstand-Beziehung-Modell:** Die Beziehungen zwischen Gegenständen (bzw. Personen) sowie deren Eigenschaften werden dargestellt. Dabei geht es nicht um konkrete Gegenstände/Personen (z.B. *die Schülerin Susi*) oder Ausprägungen von Eigenschaften (z.B. *Alter = 10 Jahre*). Stattdessen werden die Gemeinsamkeiten von ähnlichen Gegenständen oder Personen dargestellt (z.B. *Jede Schülerin und jeder Schüler hat ein bestimmtes Alter.*).
- **Klassendiagramm:** Ähnlich dem Gegenstand-Beziehung-Modell stellt das Klassendiagramm Objektgruppen und ihre Beziehungen dar. Allerdings bietet das Klassendiagramm komplexere Beziehungstypen an, z.B. *Generalisierungen* und *Teil-Ganzes-Beziehungen*. Außerdem lässt sich das Verhalten von Objekten dieser Objektgruppen ebenfalls beschreiben. Im Klassendiagramm werden die Ausprägungen der Attribute einzelner Gegenstände nicht erfasst (anders als im Objektdiagramm).



- **Aktivitätsdiagramm:** In diesem Diagramm können Abläufe, Handlungsabfolgen bzw. Handlungsanweisungen beschrieben werden. Die Darstellung ist übersichtlicher als eine textuelle Beschreibung, vor allem, wenn die Handlung kein sequentieller Ablauf ist, sondern Verzweigungen und Wiederholungen zulässt.

## Objektdiagramm

Ein Objektdiagramm stellt konkrete Objekte und ihre Eigenschaften sowie die Beziehungen zwischen Objekten dar. Jedes Objekt wird in einem rechteckigen Rahmen dargestellt. Das erste Feld beinhaltet den Namen (den Bezeichner) des Objekts. Vor dem Namen des Objekts steht eine Objektgruppe, zu der das Objekt zugeordnet werden kann. Diese beiden Angaben werden unterstrichen. Abbildung 2 zeigt ein Objektdiagramm mit zwei Objekten: Grisu und Poppy. Grisu ist ein Drache und Poppy ist ein Hund.

Im zweiten Feld werden für jedes Objekt die Eigenschaften und die Ausprägungen der Eigenschaften angegeben, ähnlich wie in einem Steckbrief. In Abbildung 2 sieht man die Eigenschaften (*Name, Alter, ...*) und ihre Ausprägungen (*Poppy, 5 Jahre ...*) für die beiden Objekte.

Zwischen einzelnen Objekten können Beziehungen beschrieben werden. Dazu werden die Objekte verbunden und die Verbindungslinie wird beschriftet. Der Pfeil gibt die Leserichtung an, z.B. *Der Drache Grisu entführt den Hund Poppy.*

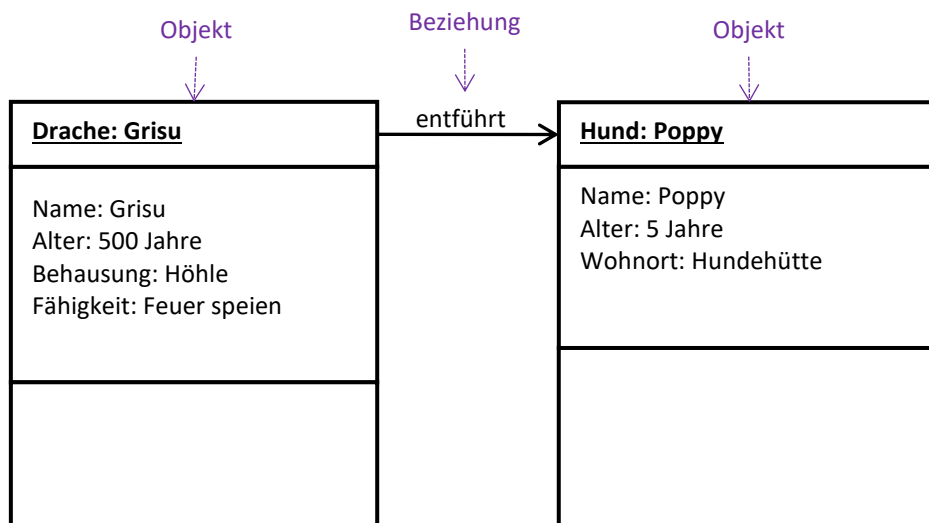


Abbildung 2: Beispiel eines Objektdiagramms

Man erstellt also (ausgefüllte) Steckbriefe von konkreten Objekten und kann die Beziehungen zwischen den Objekten darstellen.

## Gegenstand-Beziehung-Modell

Im Gegenstand-Beziehung-Modell werden Objektgruppen (z.B. die Gruppe der *Hunde*, die einen *Namen* und einen *Wohnort* haben) und keine konkreten Objekte (z.B. *Hund Poppy* mit dem *Namen Poppy* und dem Wohnort *Hundehütte*) beschrieben. Gegenstände/Personen/Objekte, die ähnliche Attribute aufweisen, werden in diesem Modell zu einer Gruppe zusammengefasst.

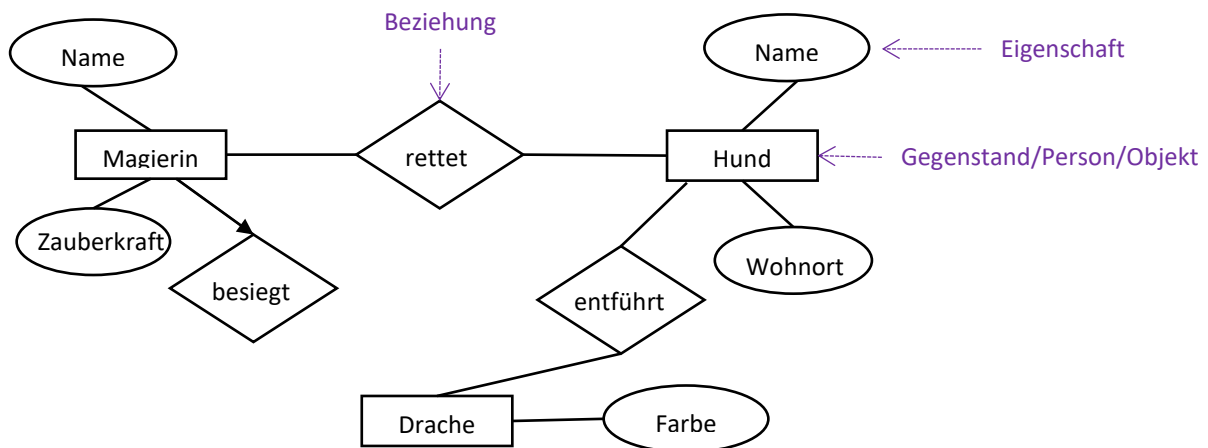


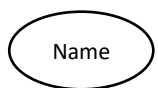
Abbildung 3: Beispiel eines Gegenstand-Beziehung-Modells

### Elemente

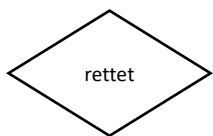
Das Gegenstand-Beziehung-Modell besteht aus drei unterschiedlichen Elementen:



**Entitätstyp (Gegenstand):** Gruppe von konkreten Objekten  
Darstellung: Rechteck



**Attribut (Eigenschaft):** Eigenschaften von einem Entitätstyp  
Darstellung: Ellipse



**Relationstyp (Beziehung):** Zusammenhang zwischen Entitätstypen  
Darstellung: Raute

Die einzelnen Elemente werden durch Linien miteinander verbunden (siehe Abbildung 3). Bei Beziehungen ist es manchmal hilfreich, die Leserichtung mit Pfeilen anzugeben (siehe Abbildung 4). Bei der Aufteilung des Diagramms ist es nicht immer möglich, dass die Leserichtung der Beziehung von links nach rechts bzw. von oben nach unten verläuft (wie im Beispiel oben). Eine Alternative ist es, das Prädikat immer so anzugeben, dass es in der gegebenen Richtung lesbar ist. Dadurch werden die Satzkonstruktionen aber umständlicher (z.B. Ein Hund wird gerettet von einer Magierin).

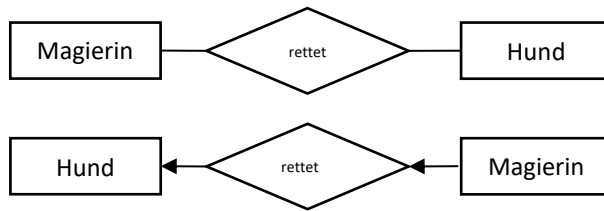


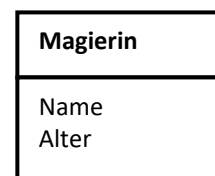
Abbildung 4: Beziehungen im Gegenstand-Beziehung-Modell

Attribute kann man mit der Erstellung eines (noch nicht ausgefüllten) Steckbriefes vergleichen: Um einen Steckbrief zu erstellen muss man sich überlegen, was man wissen will (Name, Alter, ...). Es geht nicht um das Ausfüllen des Steckbriefes (Name: Kunibert, Alter: 27).

### Übergang zum Klassendiagramm

Aus einem fertigen Gegenstand-Beziehung-Modell lässt sich leicht ein Klassendiagramm erstellen. Dieses ist übersichtlicher, da die Entitätstypen und Attribute in einem Rechteck zusammengefasst werden und nur mehr die Beziehungen der Entitätstypen untereinander durch Pfeile dargestellt werden müssen. Prinzipiell können aber alle Informationen aus den Gegenstand-Beziehung-Modell auch im Klassendiagramm dargestellt werden. Das Klassendiagramm bietet aber zusätzliche Möglichkeiten (z.B. können unterschiedliche Beziehungstypen verwendet oder das Verhalten der Entitäten ergänzt werden). Details dazu findet man im Abschnitt zum Klassendiagramm.

Im ersten Feld des Klassendiagramms wird der Entitätsname übernommen. Im zweiten Feld werden die Attribute untereinander aufgezählt. Die Beziehungen zwischen den Entitäten bleiben erhalten.



## Klassendiagramm

Das Klassendiagramm stellt (wie das Gegenstand-Beziehung-Modell) die Struktur einer Gruppe („Klasse“) von Objekten und deren Beziehungen zu anderen Gruppen dar. Zusätzlich können auch noch Informationen über das Verhalten von Objekten dieser Gruppe gegeben werden.

### Struktur

Für die Struktur werden dieselben Elemente wie im Gegenstand-Beziehung-Modell verwendet, nur die Notation ist unterschiedlich:

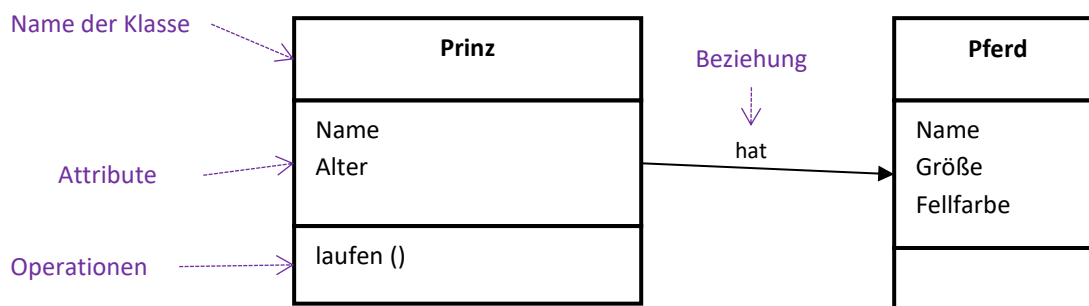


Abbildung 5: Beispiel eines Klassendiagramms

### Beziehungen

Die unterschiedlichen Beziehungen im Klassendiagramm werden durch Verbindungen der Klassen gekennzeichnet. Es gibt mehrere Beziehungstypen, die unterschiedlich dargestellt werden.

- **Assoziation:** Man möchte eine Beziehung zwischen zwei (oder mehr) Klassen herstellen. Die Verbindung erfolgt durch einen Pfeil, der auch die Leserichtung angibt. Jede Assoziation muss mit einer Bezeichnung versehen werden, die die Beziehung näher beschreibt (z.B. „hat“, „kennt“, „mag“). Im Beispiel in Abbildung 5 liest man z.B. „Ein Prinz hat ein Pferd“.
- **Generalisierung:** Diese Beziehung wird immer dann benutzt, wenn man zwei Klassen miteinander in Beziehung setzen möchte, von denen eine Generalisierung der anderen Klasse ist. Im Beispiel sind sowohl alle Hunde als auch alle Katzen Tiere, also ist die Klasse *Tier* eine Generalisierung der Klassen *Hund* und *Katze*. Die Attribute und Operationen, die bereits in der Klasse *Tier* vorhanden sind, müssen in den spezielleren Klassen *Hund* und *Katze* nicht wiederholt werden. Sie werden von der Klasse *Tier* geerbt (dh. jeder *Hund* und jede *Katze* haben einen *Namen*, auch wenn

das in ihren Klassen nicht mehr erwähnt wird). Man kann aber in den spezielleren Klassen weitere Attribute und Operationen ergänzen.

Diese Beziehung wird mit einem unausgefüllten Pfeil dargestellt, der in Richtung der generalisierten Klasse zeigt (siehe Abbildung 6):  $\longrightarrow$

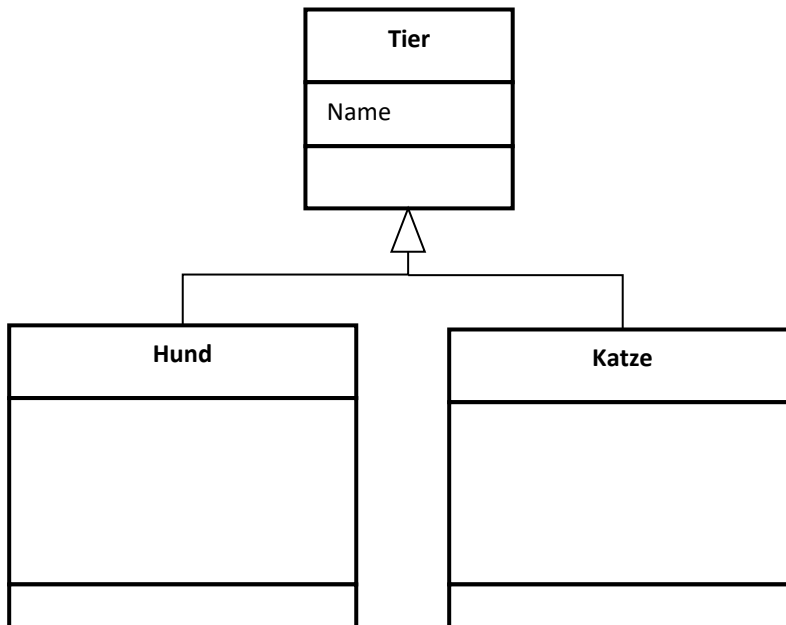


Abbildung 6: Klassendiagramm mit Generalisierung

- **Komposition:** Die Komposition stellt die Beziehungen zwischen Teilen und einem Ganzen her. Man kann also beschreiben, aus welchen Teilen eine Klasse besteht. Ein Schloss (das Ganze) besteht z.B. aus Räumen (den Teilen). Wenn (wie in diesem Beispiel) die Teile nicht ohne das Ganze existieren (ohne das Gebäude gibt es auch keine Räume), dann ist die Beziehung eine Komposition. Ansonsten ist die Beziehung eine Aggregation.

Die Komposition wird dargestellt durch eine Verbindungslinie, die an der Seite des Ganzen in einer gefüllten Raute endet.  $\blacklozenge$

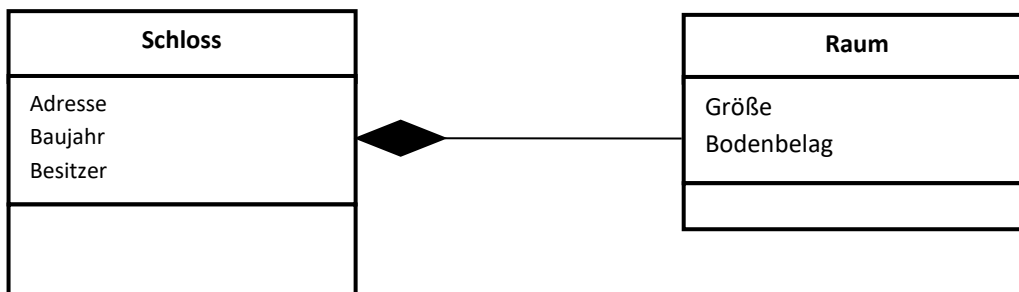


Abbildung 7: Klassendiagramm mit Komposition

- **Aggregation:** Die Aggregation stellt die Beziehung zwischen Teilen und einem Ganzen her, wobei die Teile auch ohne das Ganze existieren können. Die Rüstung besteht aus Helm, Visier und andern Bestandteilen. Der Helm und das Visier existieren aber auch ohne sie zu einer Rüstung zusammenzufassen.



Die Aggregation wird dargestellt durch eine Verbindungslinie, die an der Seite des Ganzen in einer ungefüllten Raute endet.

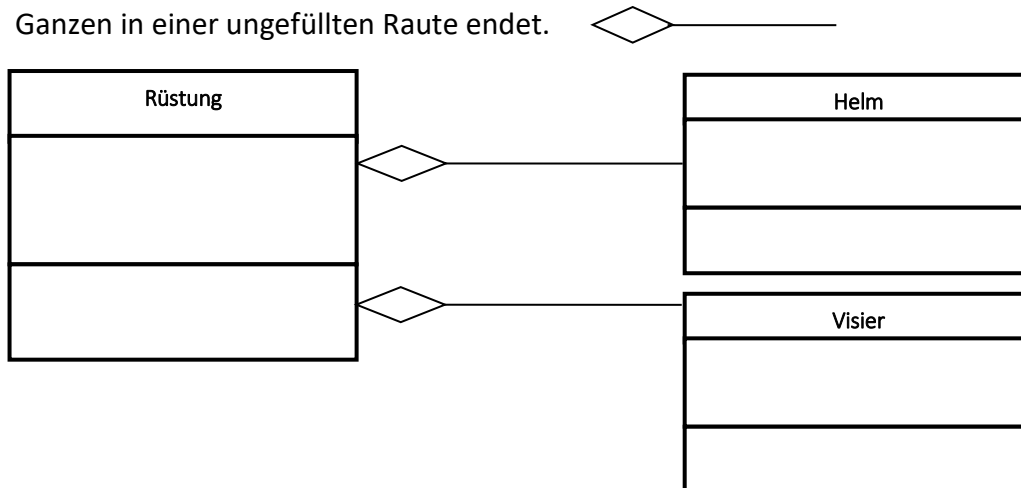


Abbildung 8: Klassendiagramm mit Aggregation

**Verhalten**

Im untersten Feld können Operationen (Methoden), die von diesen Klassen durchgeführt werden können, notiert werden (z.B. kann eine Person sprechen oder gehen). In den Klammern können Argumente (Parameter) festgelegt werden, die Informationen liefern, die für die Operation benötigt werden. Damit die Person z.B. weiß, wie weit sie gehen soll, muss die Distanz angegeben werden. Wenn man keine Argumente benötigt, lässt man die Klammern leer.

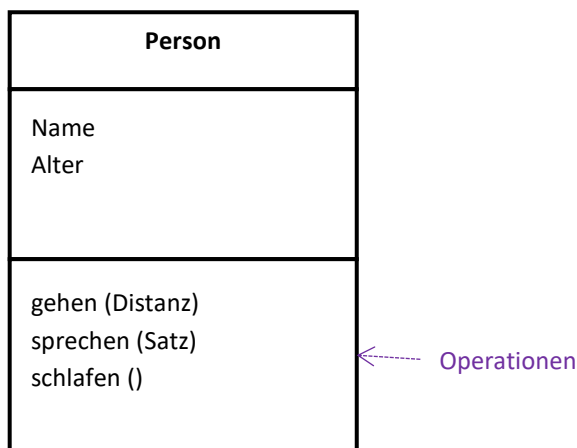


Abbildung 9: Klassendiagramm mit Methoden

Bei einer Generalisierung werden auch die Operationen vererbt. Die Klasse *Magierin* hat also nicht nur die Methode *zaubern()* sondern auch die Methoden *gehen (Distanz)*, *schlafen ()* und *sprechen (Satz)* der Klasse *Person*.

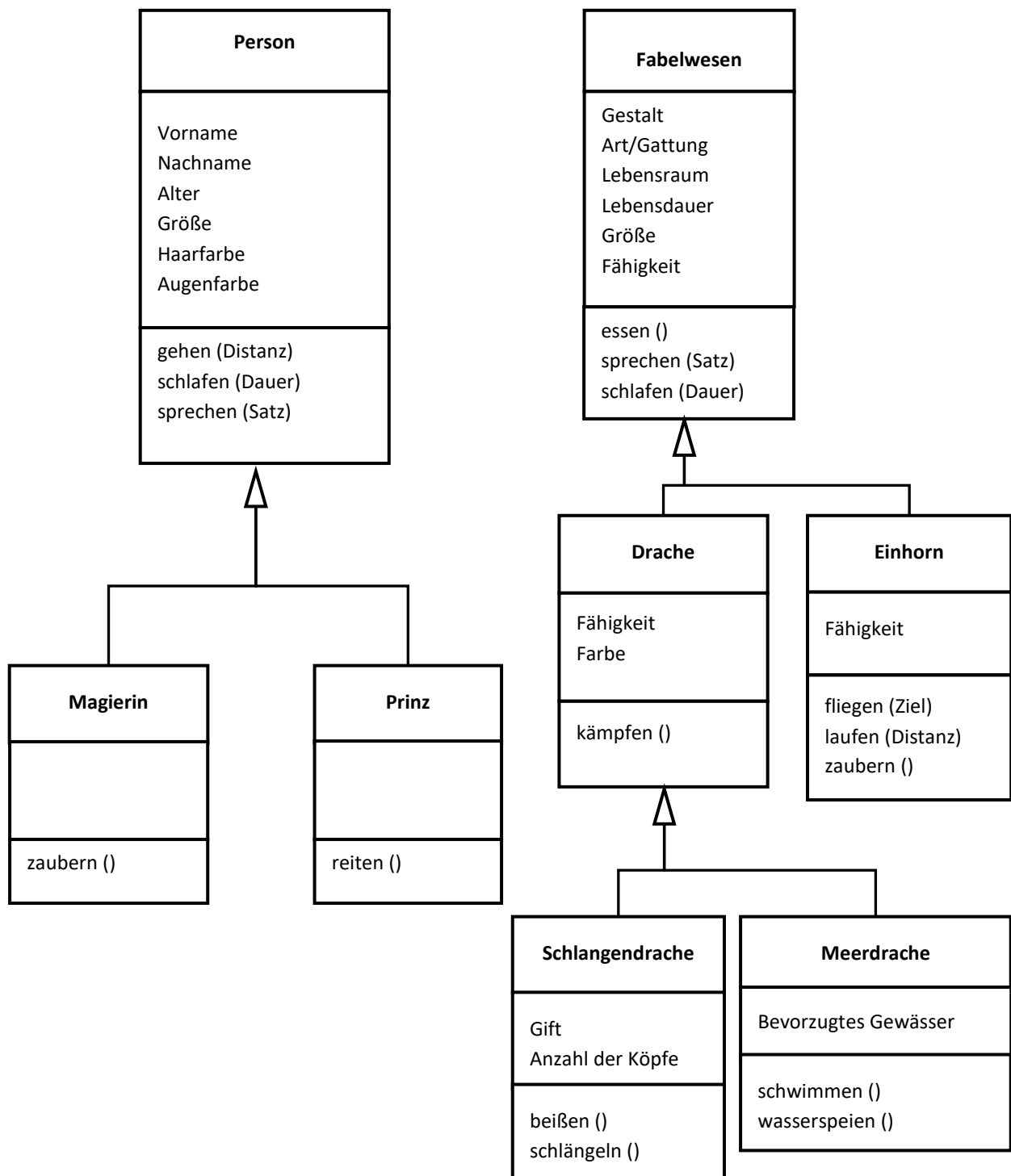


Abbildung 10: Beispiel Klassendiagramm mit Generalisierungen



## Übergang zum Objektdiagramm

Wenn man die Struktur einer Gruppe/Klasse erfasst hat, kann man sich auch mit konkreten Objekten dieser Klasse befassen. Man hat z.B. die Klasse *Person* erstellt und möchte jetzt konkrete Personen erfinden, um sie in einer Geschichte zu benutzen.

Man kann von jeder Klasse eine oder mehrere Instanzen erstellen, also Objekte der Klasse, die konkrete Eigenschaften haben. Dazu vergibt man einen Bezeichner/Namen (z.B. *Robert*) und belegt die Attribute der Klasse mit konkreten Eigenschaften (z.B. Alter: 12).

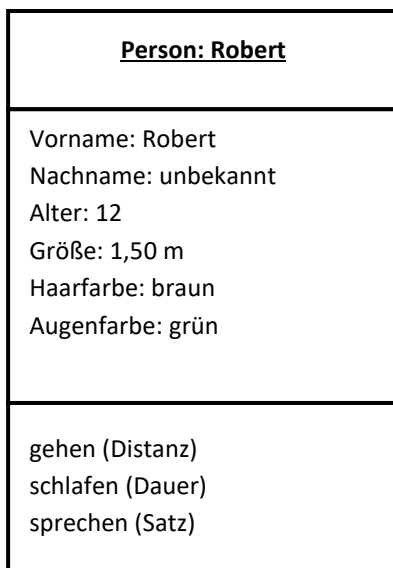


Abbildung 11: Beispiel eines Klassendiagramms

## Aktivitätsdiagramm

Das Aktivitätsdiagramm beschreibt Abläufe. Diese Abläufe werden in einzelne Aktivitäten zerlegt (jede Aktivität erhält ein eigenes Feld), deren Reihenfolge durch Pfeile angegeben werden. Eine Aktivität ist z.B.: *Poppy verschwindet*.

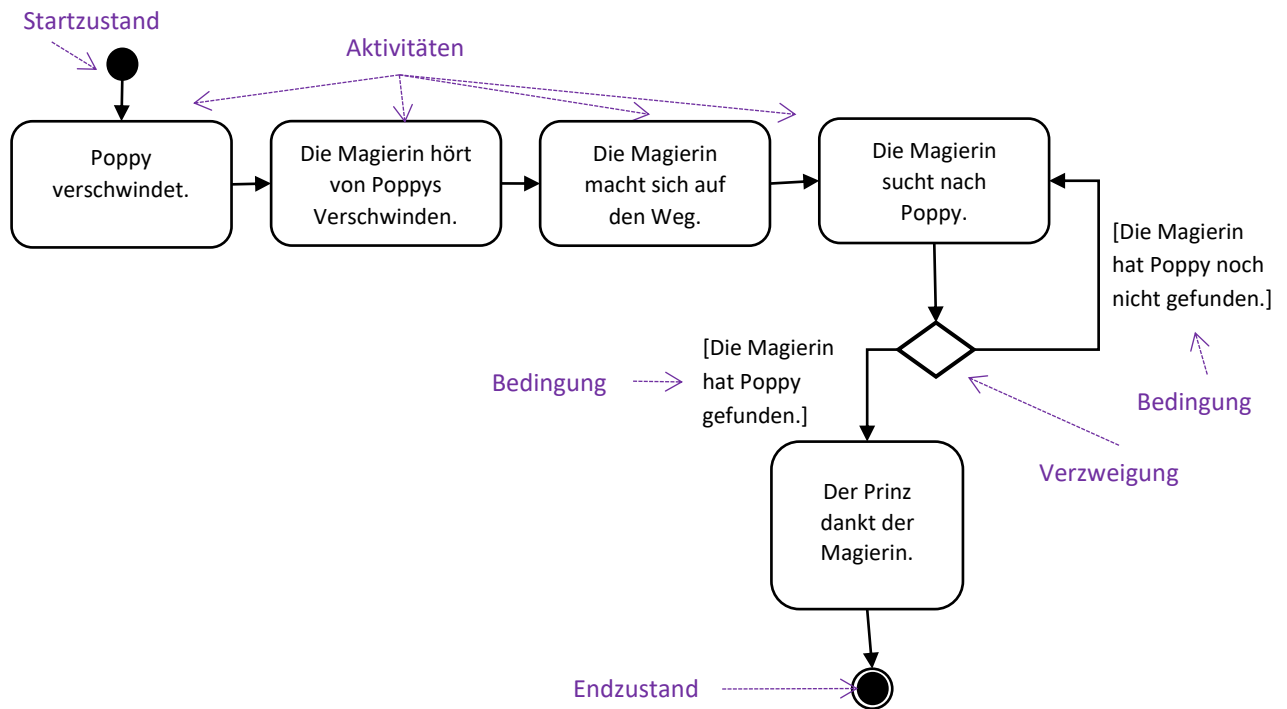


Abbildung 12: Beispiel eines Aktivitätsdiagramms

Jedes Aktivitätsdiagramm hat einen Startzustand und einen Endzustand, die Anfang und Ende markieren. Die Aktivitäten werden zwischen diesen Zuständen angeordnet.

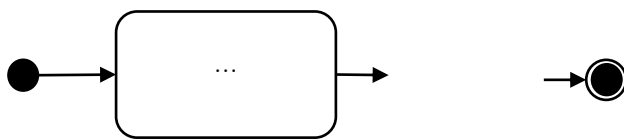


Abbildung 13: Start- und Endzustände im Aktivitätsdiagramm

## Sequenz

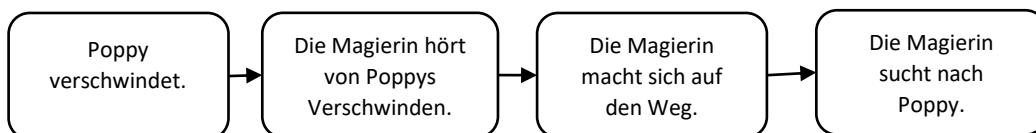


Abbildung 14: Sequenz in einem Aktivitätsdiagramm

Die Aktivitäten folgen ohne Verzweigung aufeinander. Auf jede Aktivität kann nur eine weitere Aktivität folgen. Diese werden durch Pfeile verbunden, die die Leserichtung angeben.

## Verzweigung

Wenn auf eine Aktivität mehr als eine Aktivität folgen kann, verzweigt sich der Ablauf. Man muss eine Entscheidung treffen, welchem Weg man folgt. Die Entscheidung wird dadurch getroffen, dass man prüft, welche der Bedingungen erfüllt ist, z.B. geht die Geschichte anders weiter, je nachdem, ob die Magierin Poppy bereits gefunden hat oder nicht. Um eine Verzweigung zu zeichnen, endet eine Verbindung in einer Raute ( $\diamond$ ), von der mehrere Verbindungen ausgehen können, von denen jede mit einer Bedingung versehen werden muss.

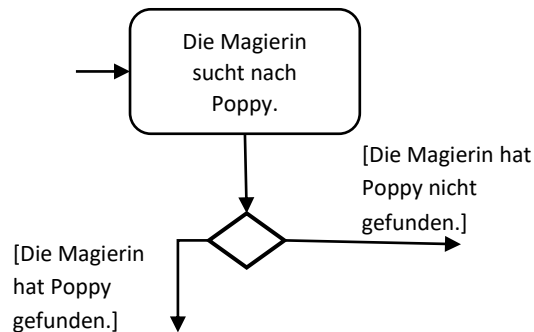


Abbildung 15: Verzweigung in einem Aktivitätsdiagramm

## Schleife

Es kann passieren, dass eine der Verbindungen in einer bereits geschehenen Aktivität endet und diese Aktivität wiederholt werden muss. Wenn Poppy noch nicht gefunden wurde, muss die Magierin z.B. weitersuchen. Die Aktivitäten in der Schleife müssen solange wiederholt werden, bis eine andere Bedingung erfüllt ist, die aus der Schleife führt. Die Magierin sucht etwa solange nach Poppy, bis die Bedingung *Die Magierin hat Poppy gefunden* eintritt.

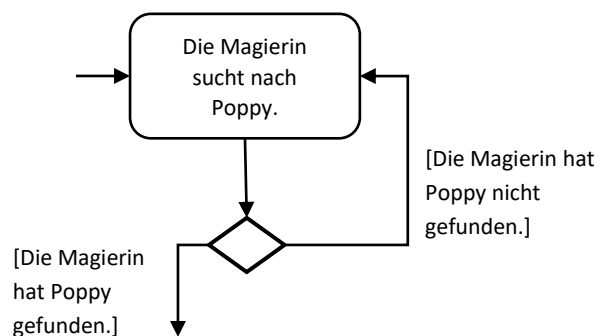


Abbildung 16: Schleife in einem Aktivitätsdiagramm