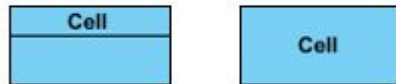


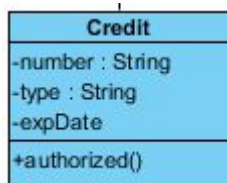
Übungsgruppe: Qianli Wang und Nazar Sopiha

Aufgabe 3-1:

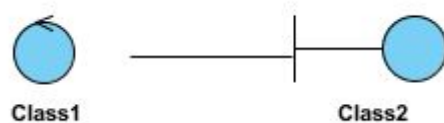
- a) **Klassen und Objekte** repräsentieren Sachen (oder Leute) aus der Realität in dem Modell. Klassen beschreibt Type von **Objekten** und gleichzeitig sind Objekte benutzbare Instanze von Klassen.



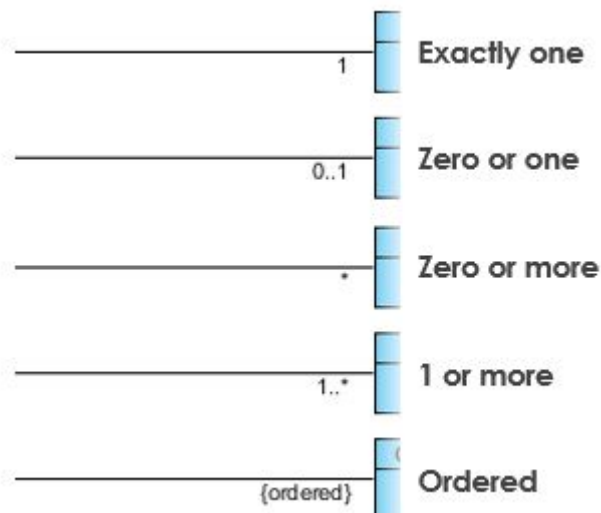
Attribute und Operationen sind Eigenschaften jedes **Objekts** und dienen zur Personalisierung und Operation sind von Klassen angebotene Dienstleistungen.



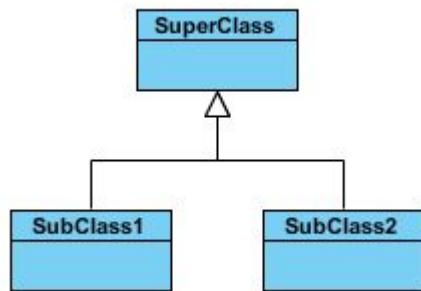
Assoziationen sind Verbindungen zwischen Entitäten (**Klassen und Objekte**)



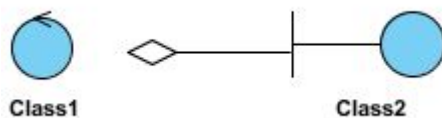
Multiplizität ist ein Hinweis darauf, wie viele **Objekte** an der angegebenen Beziehung teilnehmen können oder wie viele Instanzen des Elements zulässig sind.



Generalisierung stellt Vererbung dar (ist eine taxonomische Beziehung zwischen einem allgemeineren Klassifikator und einem spezifischen Klassifikator)



Aggregation repräsentiert einen "Teil der" Beziehung. (Klasse 2 ist ein Teil von der Klasse 1). Separate Lebenszeiten



Komposition ist wie **Aggregation**, aber die Lebenszeiten sind gleich (zur Lebenszeit der Klasse 1), wobei Teile (Klasse 2) verschwunden sind, wenn das Ganze verschwunden ist.



(Quelle: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/>)

b) Bei Attributen und Operationen, Klassen und Objekte

Notwendig:

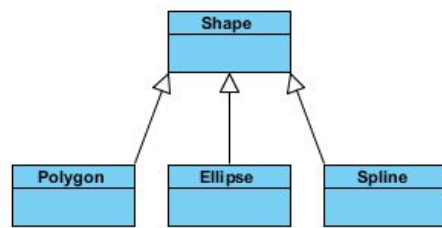
Klassenamen, Objektnamen, Attributennamen, Operationennamen.

Optional:

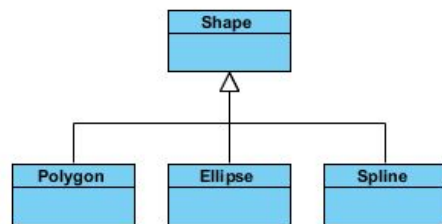
Attributen und Operationen, Visibilität, Parameter, Signatur, Typen von Attributen, Rückgabetypen von Operationen, Direktionalität

Assoziation: nichts notwendig, Name optional

Multiplizität: es gibt verschiedene Schreibweisen und Notationen. Eine der Schreibweisen ist notwendig, aber welche - optional.



Style 1: Separate target



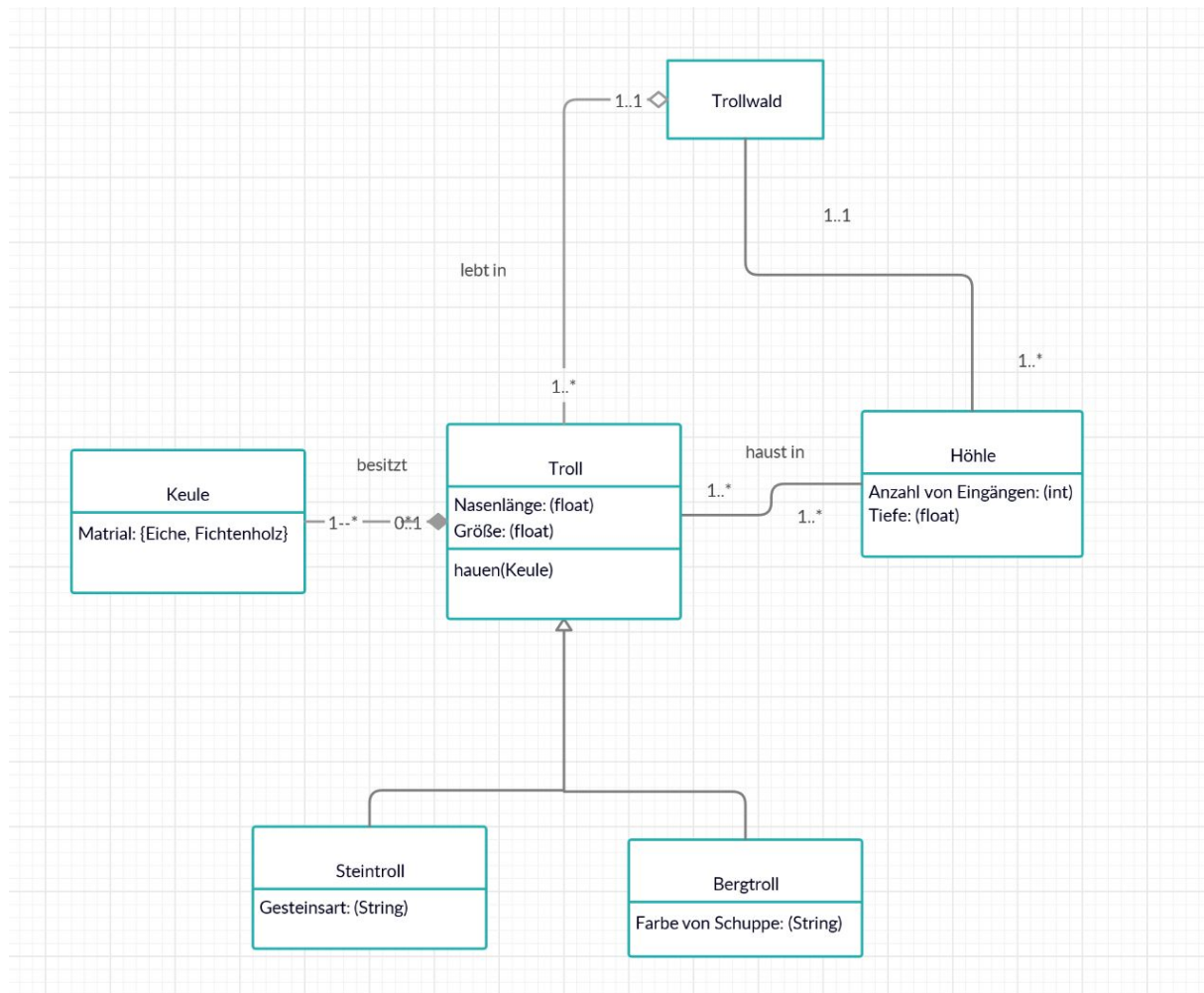
Style 2: Shared target

Generalisierung: getrenntes oder geteiltes Ziel, eine davon notwendig. Mann darf von mehreren Objekten erben.

Aggregation: nichts notwendiges

Komposition: nichts Notwendiges

Aufgabe 3-2:



Aufgabe 3-3:

a)

UML-Modellelement	Name der Metaklasse	Relevante Abschnitte
Klasse	Class	11.4 und 11.8.3
Objekt	InstanceSpecification	9.8 und 9.9.9
Generalisierung	Generalization	9.9.7
Attribut	Property	9.5 und 9.9.7
Operation	Operation	9.9.11
Komposition & Aggregation	Aggregation	11.8.1
Schnittstelle	Interface	10.5.5
Implementierung (einer Schnittstelle)	InterfaceRealization	10.5.6
Assoziation	Association	11.8.1

b)

Modellelement	Diagrammtyp	Metaklasse	Relevante Abschnitte
Methodenaufruf	Sequenzdiagramm	ActionExecutionSepecification	17.12
Interner Übergang	Zustandsdiagramm	Transition	14.5.11 und 14.2.3.8
Externer Übergang	Zustandsdiagramm	Transistion	14.5.11 und 14.2.3.8

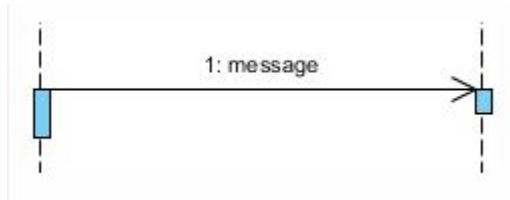
c) Aktionen im Aktivitätsdiagramm:

1. **ClearVariableAction(Variable Action) (16.14.14)**: Entfernt Wert von einer Variable.
2. **CreateObjectAction(Object Action) (16.14.18)**: Erschafft ein Objekt des angegebenen Klassifizierers.
3. **Reduce Action(16.14.45)**: Reduziert eine Sammlung durch wiederholte Kombination der Elemente.

Aufgabe 3-4:

- 1) **Sequenzdiagramm** ist ein Verhaltensdiagramm, welches die Interaktionen darstellen und beschreibt den Austausch von Nachrichten zwischen Objekten mittels Lebenslinien. Sequenzdiagramme sind zeitlich fokussiert und zeigen die Reihenfolge der Interaktion visuell an, indem sie die vertikale Achse des Diagramms verwenden, um darzustellen wann welche Nachrichten gesendet werden.
- 2) Object, Message, Attribute

- 3) Die vertikale Achse des Diagramms stellt Zeit bzw. Fortschritt dar.
- 4) Call message is a kind of message that represents an invocation of operation of target lifeline.



- 5) Nein, nicht immer. Es gibt z.B. "return message", die einfach Rückgabewert liefert und muss nicht eine weitere Methode aufrufen
- 6) Entweder wurde das Objekt von Sender erstellt(wie bei 4 im Beispiel) oder das Objekt existiert bereits und ist der Teil der Sequenz. In der Praxis weiß man ja, wie die Objekte heißen und kann sie demnach aufrufen, wenn sie public sind.
- 7) Die Zeit läuft von oben nach unten ab. Die an der Operation beteiligten Objekte werden von links nach rechts aufgelistet, je nachdem, wann sie an der Nachrichtensequenz teilnehmen. Diese Notation kann nur eine Interpretation haben
- 8) 1) Diagramname
2) Anonymous object
3) Actor
4) Create message name
5) Named object (name "tb")
6) Lifeline
7) Call message name
8) Recursive synchronous message
9) Call message (also simple, asynchronous message)
10) Synchronous message
11) Activation
12) Return message
13) Operator (opt specified on the image)
14) Condition
15) Destruction occurrence specification

(Quellen: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>
<http://agilemodeling.com/artifacts/sequenceDiagram.htm>)