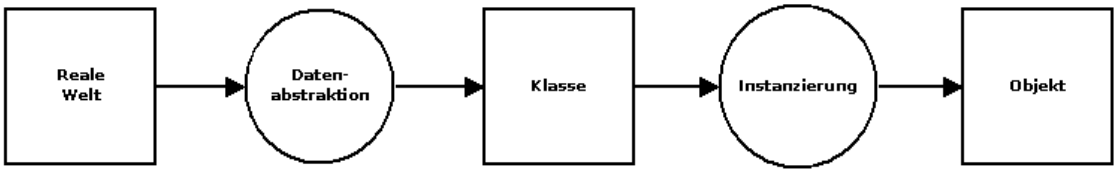
Inhaltsverzeichnis

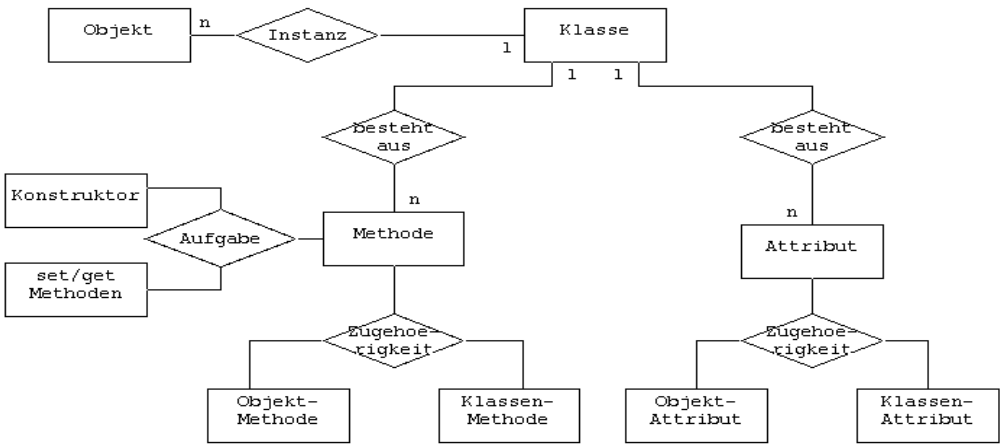
LS 1 – Infos - Grundbegriffe der Objekt-Orientierten-Programmierung

Grundbegriffe und Konzepte

- Unter **Objektorientierung** versteht man eine Sichtweise auf komplexe Systeme, bei der ein System durch das Zusammenspiel kooperierender Objekte beschrieben wird.
- Der Begriff **Objekt** ist dabei unscharf gefasst; wichtig an einem Objekt ist nur, dass ihm bestimmte **Attribute** (Eigenschaften) und **Methoden** zugeordnet sind und dass es in der Lage ist, von anderen Objekten **Nachrichten** zu empfangen beziehungsweise an diese zu senden.
- Ergänzt wird dies durch das Konzept der **Klasse**, in der Objekte aufgrund ähnlicher Eigenschaften zusammengefasst werden. Ein Objekt wird im Programmcode als **Instanz** beziehungsweise Inkarnation einer Klasse definiert.

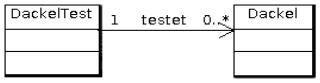


- **Kapselung** ist ein wichtiges Konzept der Objektorientierung. Als Kapselung bezeichnet man den kontrollierten Zugriff auf Methoden bzw. Attribute von Klassen. Eine Klasse hat eine Schnittstelle, die darüber bestimmt, auf welche Weise mit der Klasse interagiert werden kann.Vom Innenleben einer Klasse soll der Verwender möglichst wenig wissen müssen (Geheimnisprinzip).
- Als **Konstruktoren** und **Destruktoren** werden in der Programmierung spezielle Methoden bezeichnet, die beim Erzeugen und Auflösen von Objekten aufgerufen werden. Konstruktoren können mit Parametern versehen werden, während Destruktoren in der Regel argumentfrei sind.
- **Klassenattribute** und **Klassenmethoden** gehören zur Klasse als ganzes und nicht zu den einzelnen Objekten der Klasse.

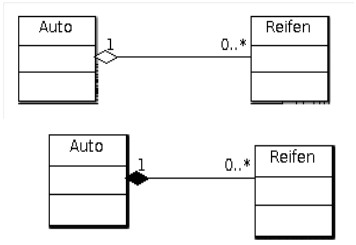


Klassenbeziehungen

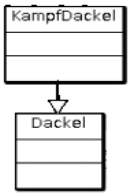
- Eine **Assoziation** ist eine allgemeine Beziehung zwischen verschiedenen Objekten einer oder mehrerer Klassen. Eine Assoziation kann genauer spezifiziert werden
  - Name
  - Rollen
  - Multiplizität (Kardinalität)
  - Richtung



- Eine besondere Form der Assoziation ist die **Aggregation**. Hierbei herrscht eine Beziehung der Art "besteht aus", wobei die Teile jedoch auch unabhängig vom Ganzen existieren können.
- Hat man eine "besteht aus" Beziehung, in der die Teile nur in Verbindung mit dem Ganzen existieren können, so spricht man von einer **Komposition**.

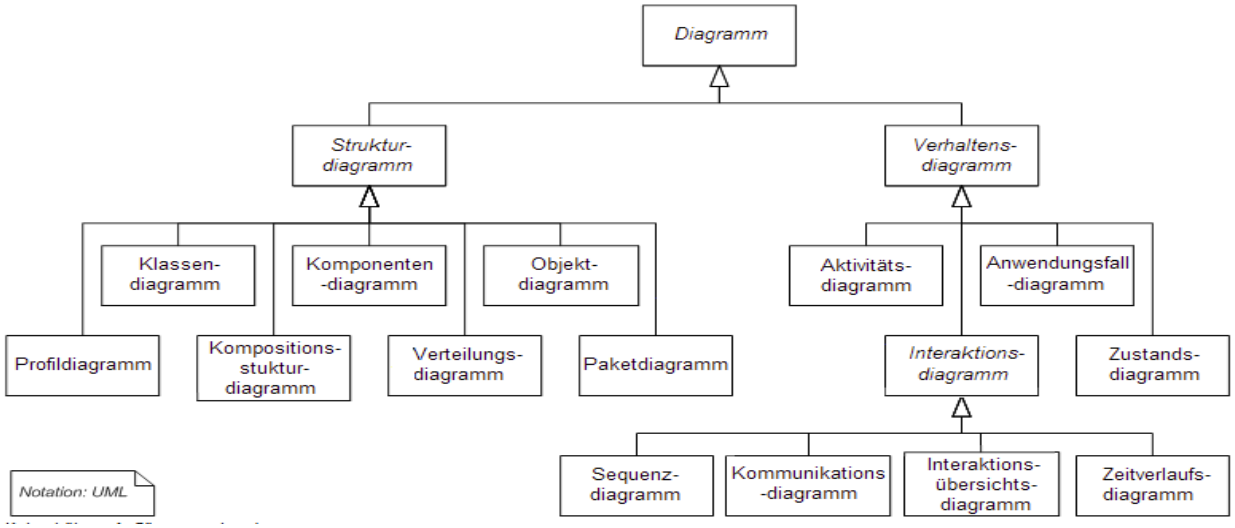


- Klassen können von anderen Klassen abgeleitet werden (**Vererbung**). Dabei erbt die Klasse (Kindklasse, Subklasse) die Attribute und die Methoden von der vererbenden Klasse (Vaterklasse, Basisklasse, Superklasse).



UML

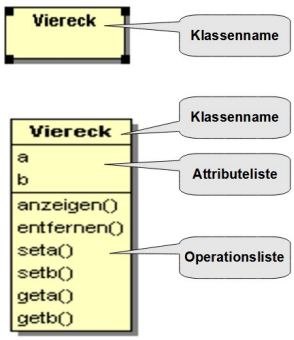
Die **Unified Modeling Language** (**UML**) ist eine von der Object Management Group (OMG) entwickelte und standardisierte Sprache (ISO/IEC 19501) für die Modellierung von Software und anderen Systemen.

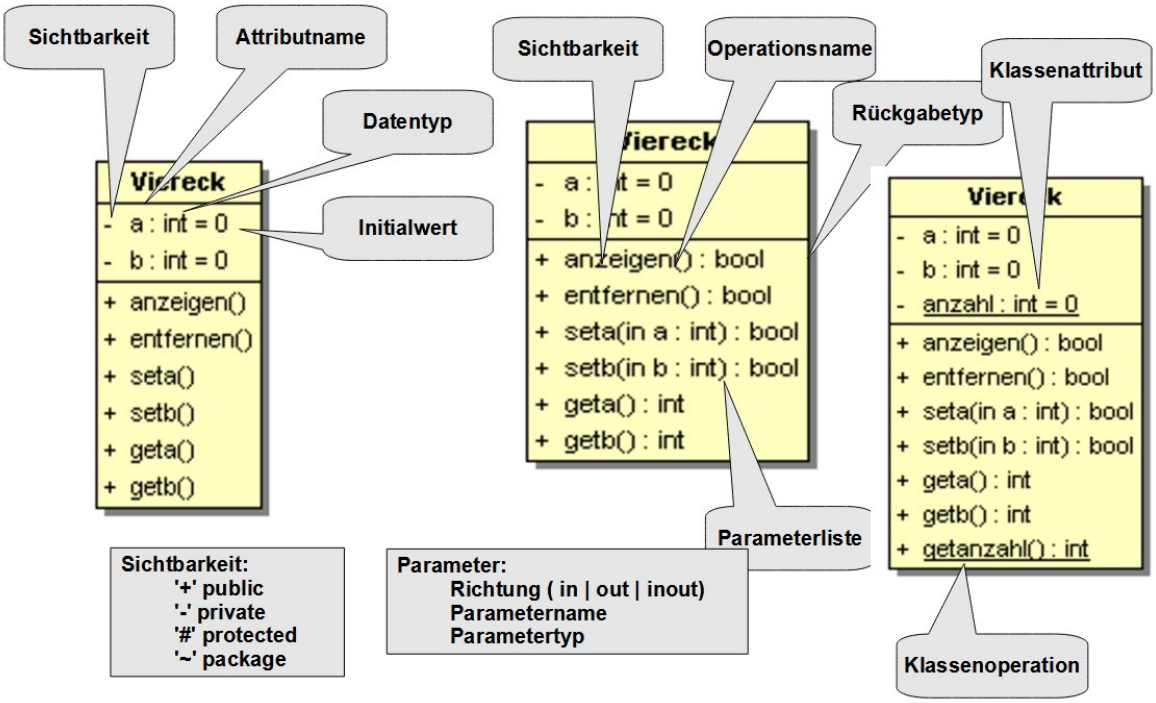


Ein **Klassendiagramm** ist ein Strukturdiagramm der UML zur grafischen Darstellung von Klassen, sowie deren Beziehungen. Typischerweise zeigt ein Klassendiagramm folgende Elemente

- Klassen
- Attribute
- Metoden
- Vererbungshierarchien
- Assoziationen und andere Beziehungen zwischen Klassen

Klassendiagramme sind der zentrale Teil der UML (wichtigster Teil)

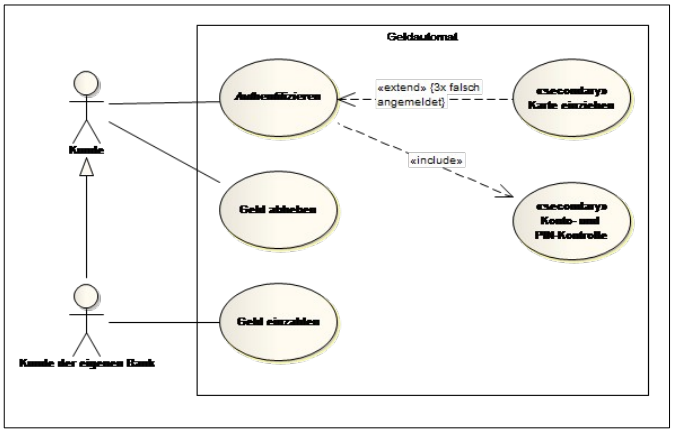




Anwendungsfalldiagramm

Verwendung:

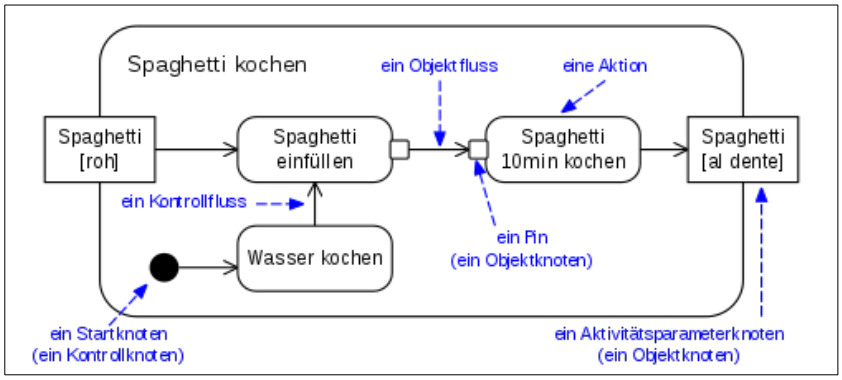
- Dokumentation von Anforderungen an das Programm (System)

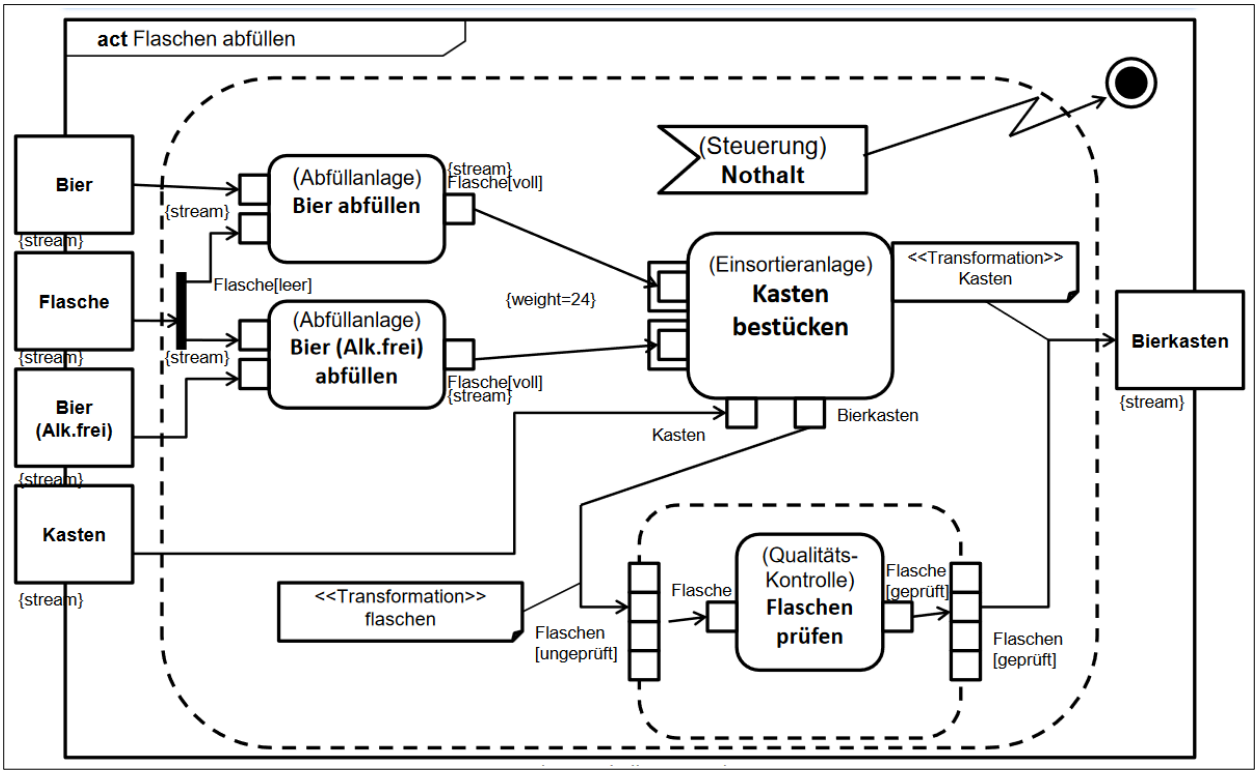


Aktivitätendiagramm

Verwendung:

- Beschreibung eines Anwendungsfalls
- Ablauf von Geschäftsprozessen und Anwendungsfällen
- Ablauf von Aktionen in einer Methode

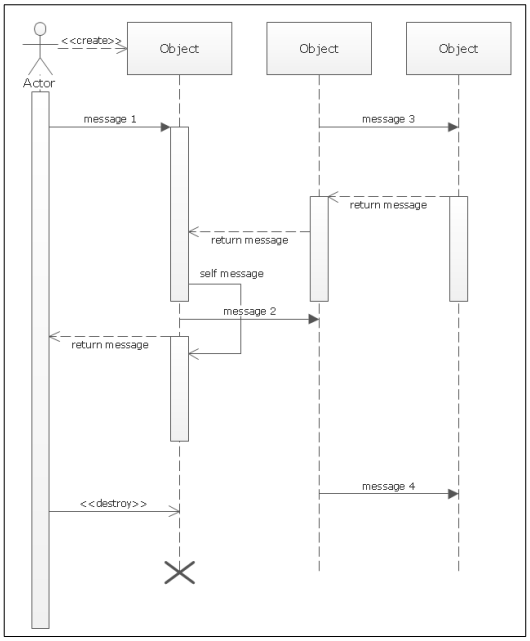




Sequenzdiagramm

Verwendung:

- Zusammenspiel verschiedener Objekte durch Methodenaufrufe (Nachrichten)



Sonstige Informationsquellen

http://www.omg.org/spec/UML/

LS 1 – Infos – Klassen für Aufgabe 2

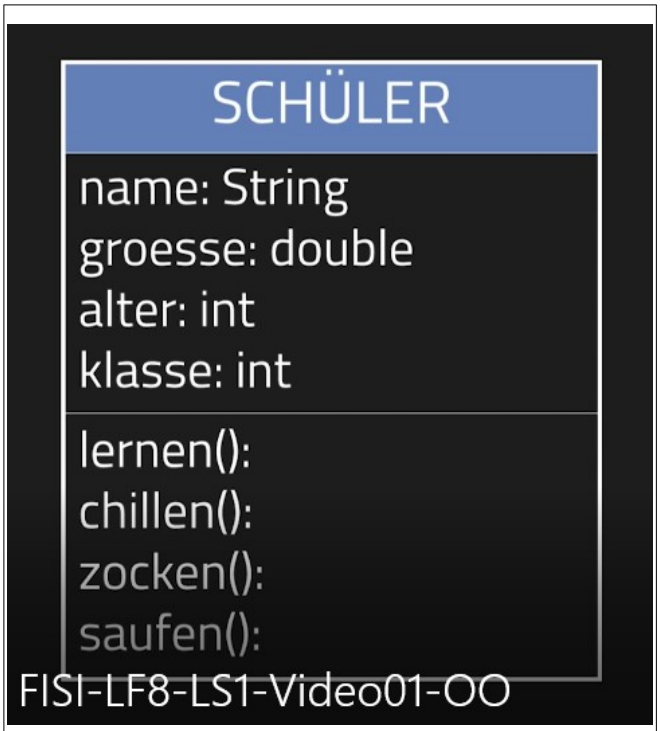


Schaubild 1: Klasse Video 1

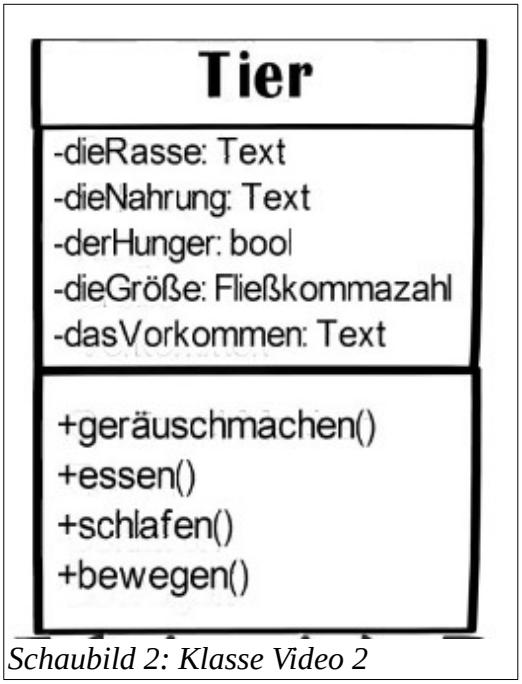
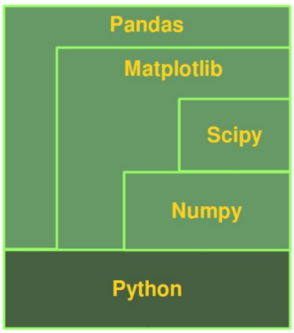


Schaubild 2: Klasse Video 2

LS 1 – Infos – Pandas

Was ist pandas?

- Python Modul zur Verwaltung und Analyse von Daten
- pandas bietet mehrere Datenstrukturen an:
  - **Dataframes**
  - Series
- Einsatzmöglichkeiten von pandas:
  - Zwischen Datenformaten umwandeln
  - Daten auswerten
  - Daten aufarbeiten



1

Wie kann man einen Dataframe anlegen?

a) mit Python-Dictionary → `DataFrame()`

Wie kann man Daten aus einer Datei in ein Dataframe einlesen bzw. Ein Dataframe in einer Datei speichern?

a) aus CSV-Datei einlesen → `read_csv()`
b) aus JSON-Datei einlesen → `read_json()`
c) aus Excel-Datei einlesen → `read_excel()`
d) in CSV-Datei speichern → `to_csv()`
e) in JSON-Datei speichern → `to_json()`
f) in Excel-Datei speichern → `to_excel()`

Wie kann man Daten aus einer Datenbank in ein Dataframe einlesen?

a) aus Datenbank einlesen → `read_sql()`

Wie kann man auf Dataframeelement zugreifen?

- Ganze Spalte adressieren → `df["Gewicht"]`
- Ganze Zeile adressieren → `df.loc[37]`
- Zelle adressieren → `df["Gewicht"][5]`

1

- → `df.at[], df.iat[], df.loc[], df.iloc[]`

Wie kann man Metadaten eines Dataframe abrufen?

a) Form des Feldes → `.shape`
b) Anzahl Dimenesionen → `.ndim`
c) Anzahl Elemente → `.size`
d) Datentypen → `.dtype`
e) Anzahl Elemente pro Spalte → `info()`

Wie kann man Daten in einem Dataframe statistisch auswerten?

a) Spaltenweise Auswertung → `count(), mean(), median(), std(), ...`
b) Spaltenübergreifende Auswertung → `corr()`

Wie kann man Daten in einem Dataframe bereinigen?

a) Zeilen mit fehlenden Daten löschen → `dropna()`
b) Fehlenden Daten ersetzen → `fillna()`
c) Falsches Datumsformat verbessern → `to_datetime()`
d) Duplikate entfernen → `drop_duplicates()`

Wo finden ich weitere Informationen zu pandas?

https://www.w3schools.com/python/pandas/default.asp

https://www.python-kurs.eu/pandas.php

https://pandas.pydata.org/