

**Lehramt Mathematik  
Lehramt Naturwissenschaften  
Fortbildung für Lehrer:innen  
„Grundlagen digitaler Lehr-Lern Szenarien“  
- Modelle und Modellierung I -**

**Prof. Dr. Sven Hofmann**

Institut für Informatik  
Professur für Didaktik der Informatik

 0341 / 97 32325

 [sven.hofmann@informatik.uni-leipzig.de](mailto:sven.hofmann@informatik.uni-leipzig.de)

# Übersicht über die Themen der Vorlesungsreihe

1. Grundlagen der Mediengestaltung
2. Computergrafik
3. Interaktive Medien
4. **Modelle und Modellierung**
5. Aspekte des e-Learning
6. Erstellen und Gestalten von Webpräsentationen

## Einleitendes Beispiel-Problem aus der Schulpraxis

Ein Praktikant (Referendariat, Blockpraktikum B) möchte unterschiedliche **Unterrichtsmodelle** ausprobieren. Er bittet Sie um Rat, mit welchen Methoden und Mitteln er diese umsetzen kann.

Während des Mathematik-Unterrichts bemerken Sie, dass der Abstraktionsgrad des vermittelten Inhalts zu hoch ist. Sie benötigen ein (vereinfachtes) **Modell**, um den Schülerinnen und Schülern den Sachverhalt zu veranschaulichen.

Die Schülerinnen und Schüler sollen selbständig ein Produkt entwickeln (z.B. ein Gedicht, ein Mindmap, ein Drahtnetzmodell eines Würfels, ...). Dabei benötigen sie Anleitung durch ein **Vorgehens-Modell** um das Problem lösen zu können.

Die Schule entschließt sich, wegen Lizenzproblemen auf eine andere Software umzusteigen (z.B. andere Tafelsoftware, MS-Office zu OpenOffice,...). Der Schulleiter bittet Sie, für die Kolleg-Innen eine „**Softwareschulung**“ durchzuführen.

# Gliederung

## 1. Modelle, Modellbegriff

- Allgemeine Merkmale
- Zweckorientierung

## 2. Modelle im der Informatik

- Modelltypen
- Arten der Modellierung

## 3. Objektorientierte Modellierung

- Grundlegende Thesen
- Darstellungsformen
- OOM als Software-Konzept

## 4. Modellierung von Rechnernetzen (Script Teil II)

- Begriffe, Client-Server-Modell,
- Kommunikation in Rechnernetzen, Rechnernetz-Protokolle
- Dienste in Rechnernetzen

# Grundlegende Merkmale von Modellen

(Modelltheorie nach Stachowiak 1973)



[www.wikipedia.org](http://www.wikipedia.org)



[www.zug-um-zug.de](http://www.zug-um-zug.de)

Ein **Modell** ist eine von einem Subjekt erstellte, vereinfachte Abbildung eines Objektes bzw. Ausschnittes der Realität.

Jedes Modell steht zu einem Objekt oder dessen mentaler Abbildung in Relation.

Ein Objekt kann durch mehrere Modelle repräsentiert werden.

# Grundlegende Merkmale von Modellen

(Modelltheorie nach Stachowiak 1973)



# Modellierung

**Zweckorientierung** der Modellbildung:

- ↳ • **Erkenntnistheoretische** Modellierung  
 → experimentieren, simulieren am Modell



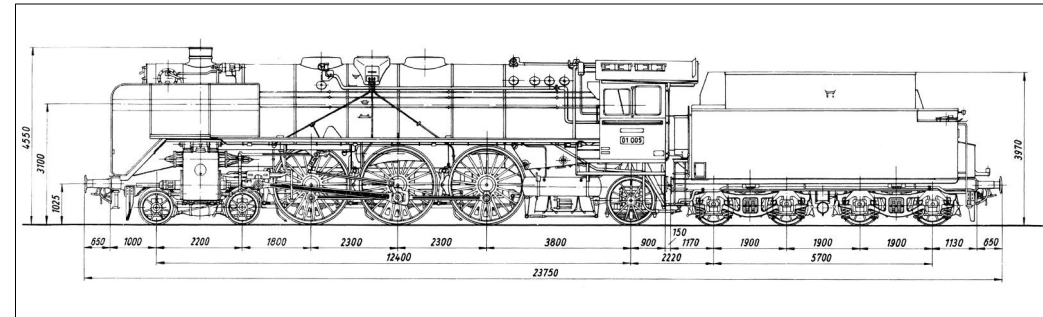
TU Dresden, Institut für Wasserbau und  
Hydromechanik ([www.tu-dresden.de](http://www.tu-dresden.de))

Für Mat.-Nat. Unterricht: z.B. <https://phet.colorado.edu/de/simulations>

# Modellierung

## Zweckorientierung der Modellbildung:

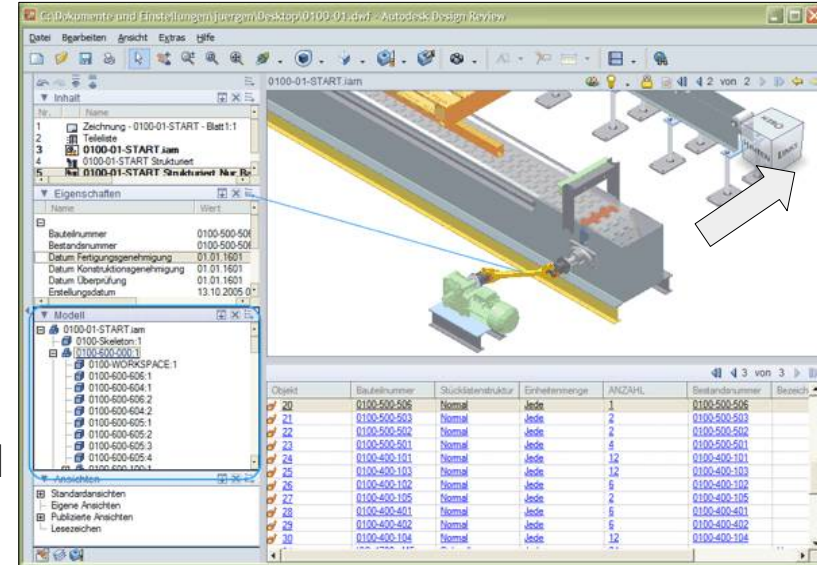
- ▶ • **Erkenntnistheoretische** Modellierung  
 → experimentieren, simulieren am Modell
- ▶ • **Technische** Modellierung  
 → Modell als Ersatz für Original



# Modellierung

## Zweckorientierung der Modellbildung:





- ➔ • **Erkenntnistheoretische** Modellierung  
→ experimentieren, simulieren am Modell
- ➔ • **Technische** Modellierung  
→ Modell als Ersatz für Original
- ➔ • **Generische** Modellierung  
→ Modell als Muster für zukünftiges Original



Autodesk Inventor  
([www.lh3.ggph.com](http://www.lh3.ggph.com))

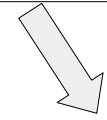
# Modellierung

## Zweckorientierung der Modellbildung:

- 
 • **Erkenntnistheoretische** Modellierung  
 → experimentieren, simulieren am Modell
- 
 • **Technische** Modellierung  
 → Modell als Ersatz für Original
- 
 • **Generische** Modellierung  
 → Modell als Muster für zukünftiges Original
- 
 • **Logische** Modellierung  
 → Abbildung des Originals als System  
 von Zeichen im Modell

```

termin(montag,logik,10,hs_g).
termin(mittwoch,logik,8,hs_g).
termin(dienstag,physik,10,w3_1_161).
termin(donnerstag,physik,10,w3_1_161).
termin(montag,gpi,14,hs_g).
termin(mittwoch,gpi,10,hs_g).
  
```



Anfrage nach Logikveranstaltung:  
?- termin(X,logik,Y,Z).

X = montag  
Y = 10  
Z = hs\_g ;

X = mittwoch  
Y = 8  
Z = hs\_g ;

# Modell-Typen in der Informatik

## Architekturmodelle

- Maschinenmodell (Turing, Automaten)
- Rechenmodell (imperativ, funktional, logisch)
- Referenzmodell (OSI, **Client-Server, Farben**)

## Vorgehensmodelle

- Phasenmodell (Wasserfallmodell)
- **Objektorientierte Modellierung** (Klassen-D., Zustands-D., **UML**)

## Entwurfsmodelle

- Systemmodell (Funktion, Daten, Zustand, Ereignis)
- **Datenbankmodell (ER-Modell, Normalform)**
- Komponentenmodell

## Untersuchungsmodelle

- Mathematisches Modell (abstrakte Modelle, Verifikation)
- Simulationsmodell (diskret, kontinuierlich)

# Arten der Modellierung in der Informatik

## Datenmodellierung

Anwendungsbeispiel: Sie erhalten von der Schulleitung die Aufgabe, die Ganztagsangebote Ihrer Schule zu verwalten.

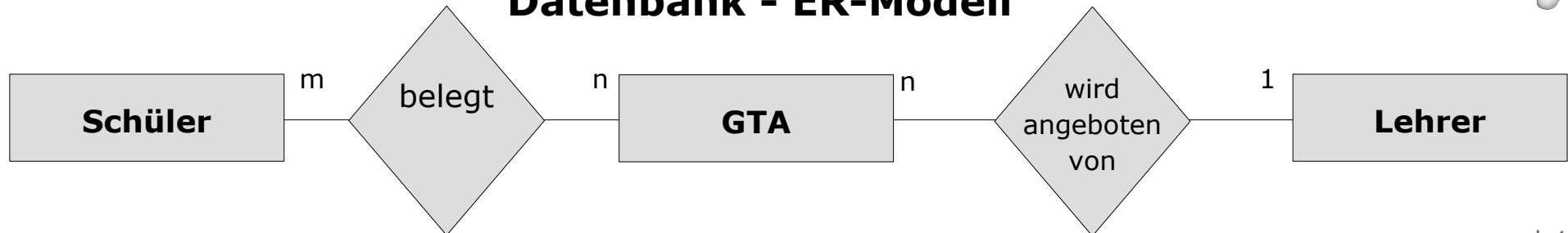
Wie modellieren Sie eine solche Verwaltung?

Sachlage: Die Schülerinnen und Schüler einer Schule können verschiedene Ganztags-Angebote belegen.

Diese GTA werden von jeweils einem GTA-Lehrer (intern/extern) gehalten.



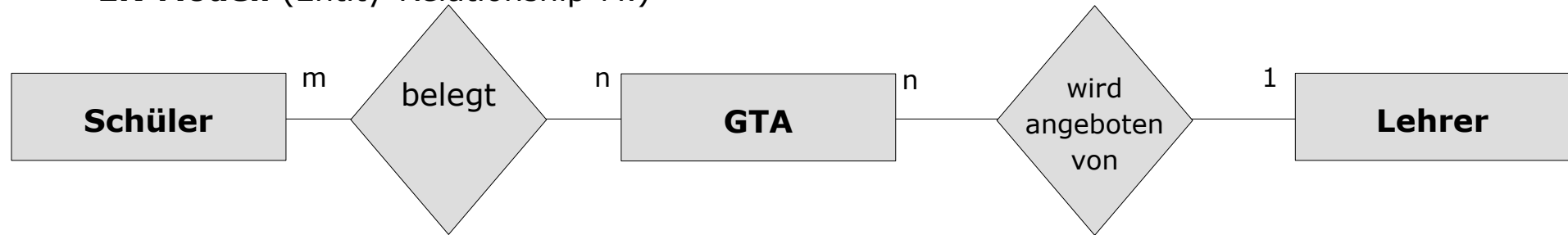
### Datenbank - ER-Modell



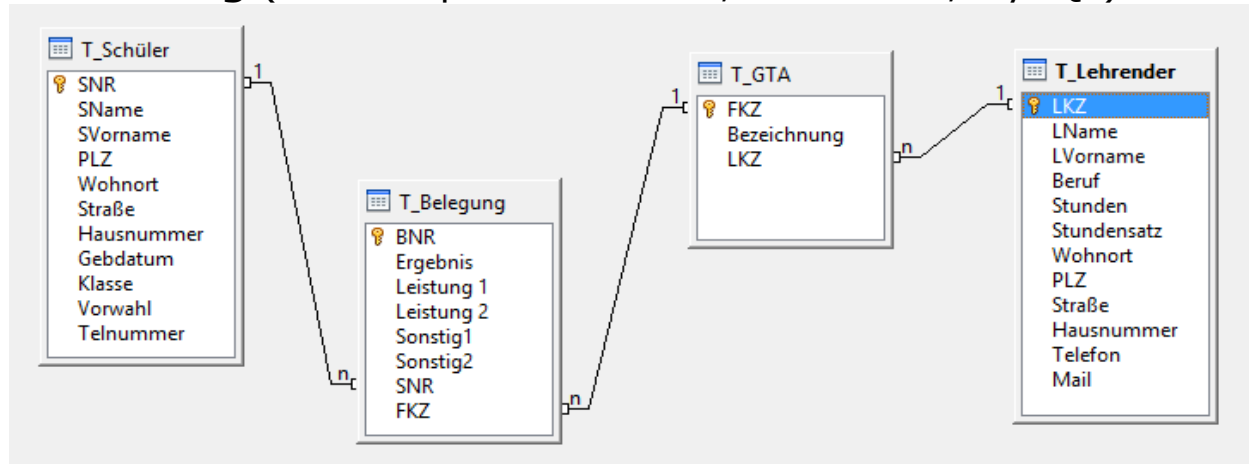
# Arten der Modellierung in der Informatik

## Datenmodellierung

ER-Modell (Entity-Relationship-M.)



Umsetzung (z.B. in OpenOffice Base, MS-Access, MySQL)

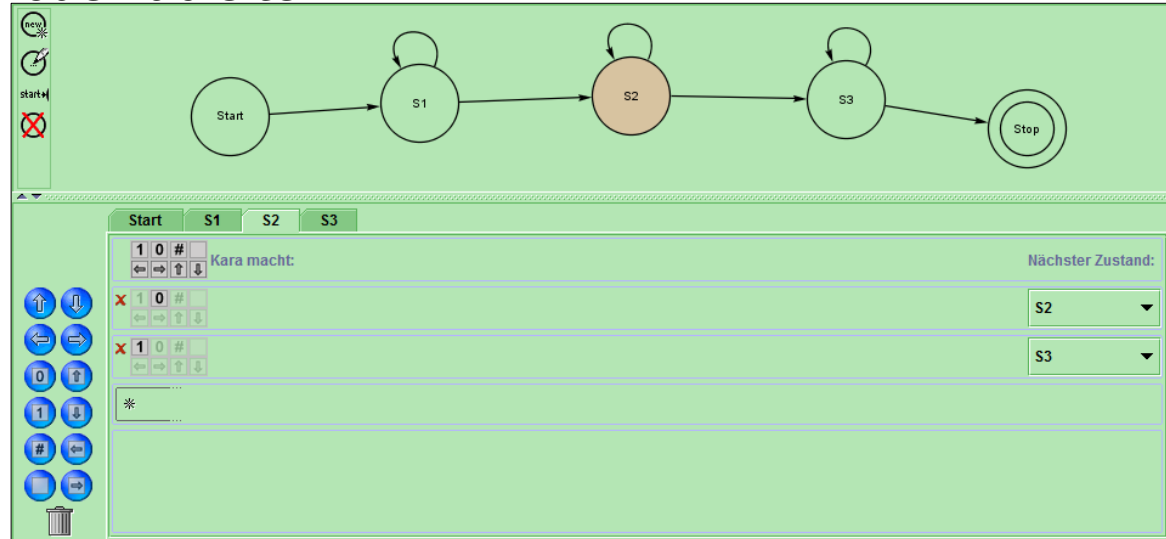


# Arten der Modellierung in der Informatik

## Zustandsmodellierung

Abläufe mit geeigneten Diagrammen darstellen, insbesondere Zustände und Übergänge  
 Beispiel: Die Schüler sollen ihren Umgang mit einem technischen Automaten beschreiben und daraus das Modell ableiten

Anwendungsbeispiel:  
 Die Eingangstür zum Lehrerzimmer ist mit einem Zahlencode gesichert.  
 Dieser „Automat“ soll nur die Eingabe der Zeichenfolge  
 0-->1-->0  
 akzeptieren.



Umsetzung (z.B. TuringKara)  
<http://swisseduc.ch/informatik/karatojava/turingkara/index.html>

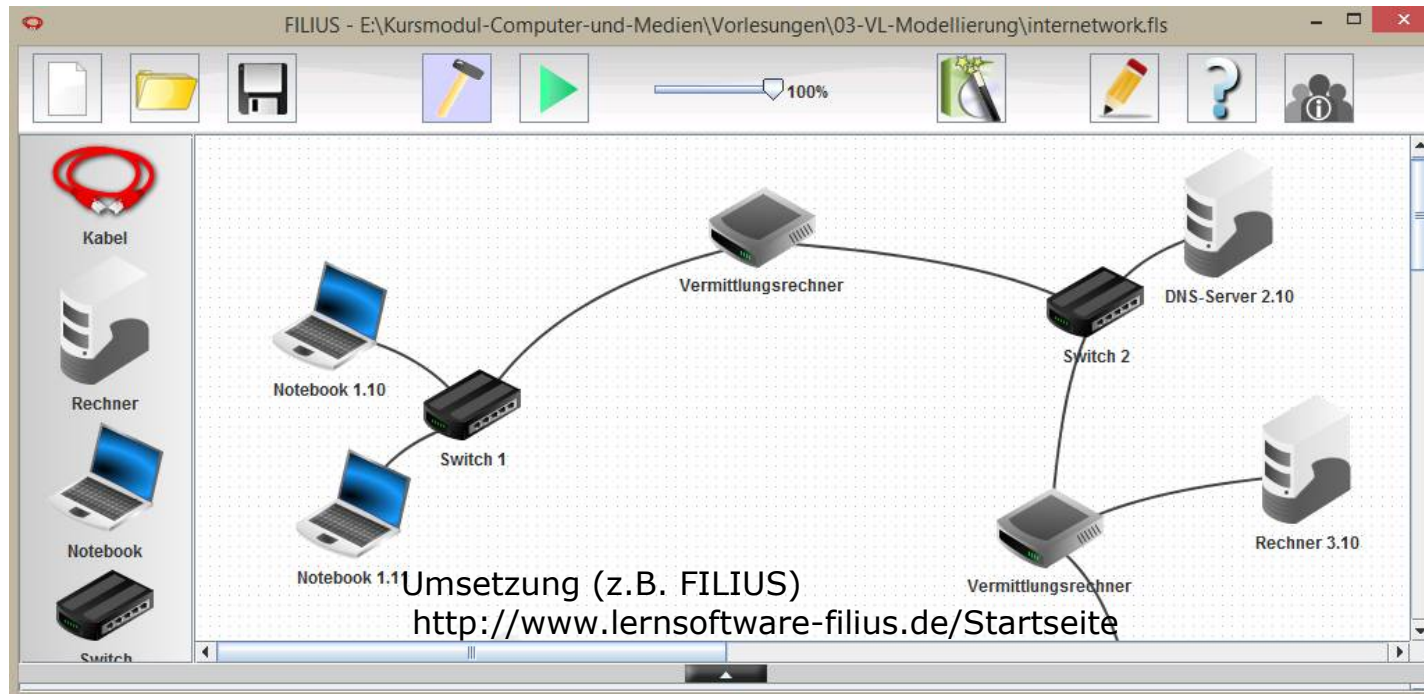
Turing Kara starten

# Arten der Modellierung in der Informatik

## Funktionale Modellierung

Komplexe Gebilde in Bausteine zerlegen, Datenfluss untersuchen

Beispiel: Modell eines Rechnernetzes



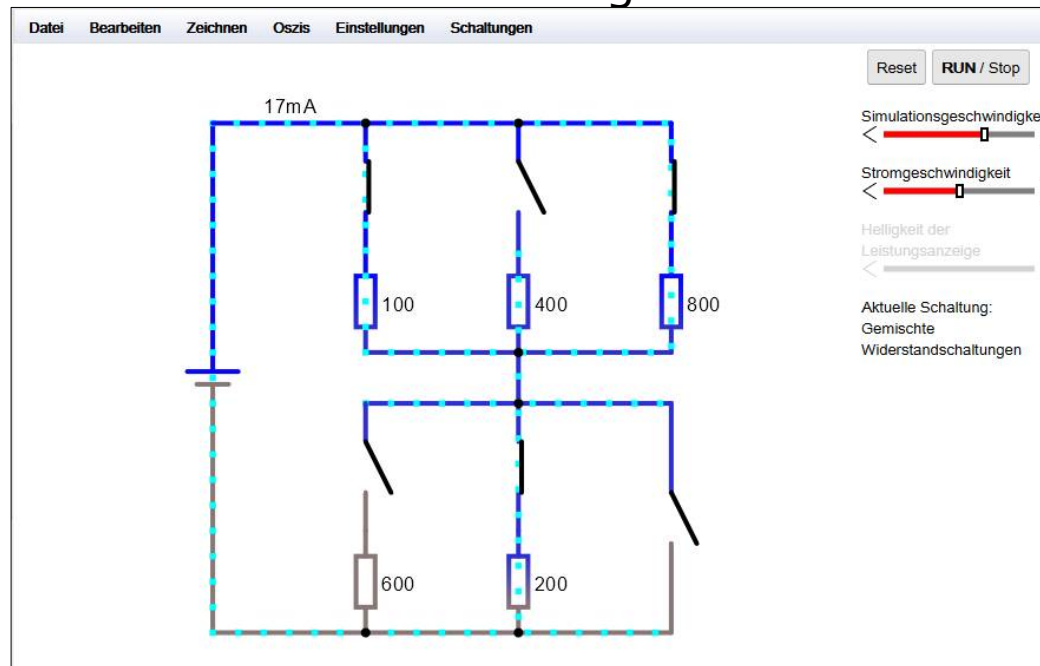
FILIUS starten

# Arten der Modellierung in der Informatik

## Funktionale Modellierung

Komplexe Gebilde in Bausteine zerlegen, Datenfluss untersuchen

Fächerverbindend: Schaltungs-Simulator für Physikunterricht



<https://www.falstad.com/circuit/circuitjs.html>

Simulator starten

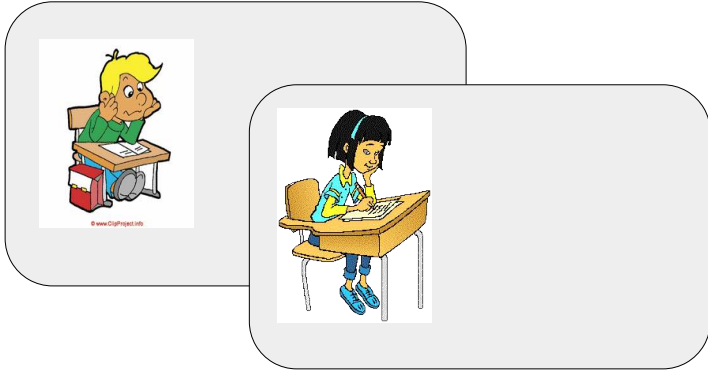


# Grundprinzipien der Objektorientierten Modellierung

- z.B. Mein\_Skoda, Omas\_Fahrrad, Monika
- Objekte besitzen `Attribute` welche durch einen Bezeichner beschrieben und mit `Werten` aus einem Wertebereich belegt sind.  
z.B. Farbe (Text), Pin (Zahl), Datum (Zahl)
- Objekte können mit `Methoden` auf Einflüsse aus ihrer Umwelt reagieren.  
z.B. motor\_starten(), fahren(), lernen()
- Objekte mit `Instanzvariablen` (aber durchaus unterschiedlichen Attributwerten) werden in einer `Klasse`  
z.B.  
Mein\_Skoda: FAHRZEUG → „Mein Skoda ist Objekt der Klasse FAHRZEUG.“  
Monika: SCHULKLASSE → „Monika ist Objekt der Klasse SCHULKLASSE.“



# Klassen und Objekte

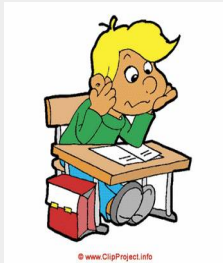


Ein **Objekt** ist eine identifizierbare, abgrenzbare Einheit der Realität oder der Vorstellungswelt. Der Zustand des Objekts wird durch Attribute mit individuellen Werten beschrieben, die Art und Weise seiner Reaktion auf Einflüsse durch Methoden.

Eine **Klasse** ist ein „Bauplan“ für eine Menge gleichartig strukturierter Objekte. Deren gemeinsame Struktur wird durch Attribute eines Wertebereiches und durch Methoden beschrieben.

# Darstellungsformen in der OOM

## Umgangssprache



Max ist ein **Objekt** mit den **Attributen**

- Name: Windowski
  - Vorname: Max
  - Geburtsjahr: 1999
- und den **Methoden**

- lesen()
- schreiben()
- zuhören()



Nina ist ein **Objekt** mit den **Attributen**

- Name: Linuxov
  - Vorname: Lina
  - Geburtsjahr: 1998
- und den **Methoden**

- lesen()
- schreiben()
- zuhören()



Die Klasse SCHUELER besitzt Objekte mit den **Attributen**

- Name: Windowski, Linuxov, Mactosch
- Vorname: Max, Lina, Herbert
- Geburtsjahr: 1999, 1998, 2000

und den **Methoden**

- lesen()
- schreiben()
- zuhören()

# Darstellungsformen in der OOM

## Klassen- / Objekt-Diagramme

### UML

Das **Objekt** schueler\_46 als  
Teil der **Klasse** SCHUELER

schueler\_46: SCHUELER

name = Windowski  
vorname = Max  
geburtsjahr = 1999

lesen(<Medium>)  
schreiben(<Dokument>)  
zuhoeren()



Die Klasse SCHUELER besitzt Objekte mit den  
**Attributen**

- Name: Windowski, Linuxov, Mactosch
- Vorname: Max, Lina, Herbert
- Geburtsjahr: 1999, 1998, 2000

und den **Methoden**

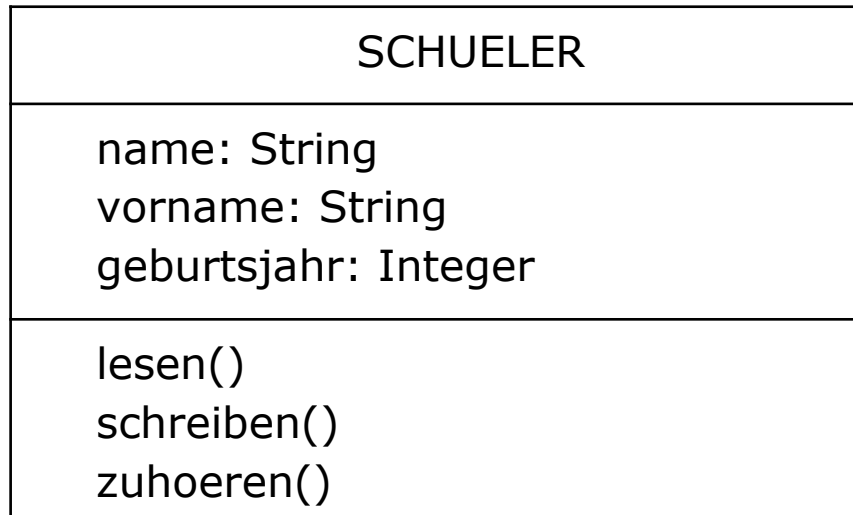
- lesen()
- schreiben()
- zuhören()

UML: **U**nified **M**odeling **L**anguage

# Darstellungsformen in der OOM

## UML - Unified Modelling Language

Wichtige Konvention: Bezeichner (Namen, **auch Dateinamen!**) bestehen grundsätzlich aus Ziffern und/oder Buchstaben OHNE Leerzeichen und OHNE deutsche Umlaute!



```

public class SCHUELER extends JAHRGANGSSTUFE {
    public String name;
    public String vorname;
    public Integer geburtsjahr;

    public void lesen() { }
    public void schreiben() { }
    public void zuhoeren() { }
}
  
```

Die **Klasse** SCHUELER  
in UML-Notation

Generieren des  
Programmcodes

# OOM als Softwarekonzept

## Basis-Konzept für Präsentations-Software

Das unter dem Dateinamen  
*uebung1.odp* gespeicherte  
Dokument ist ein Objekt der  
Klasse PRAESENTATION



uebung1.odp: PRAESENTATION
autor= Hans Kraftpunkt titel= Interaktiv Lernen datum= 15.12.2020
praesentation_erzeugen() praesentation_oeffnen() praesentation_speichern()

Das Objekt *gliederung1*  
ist ein Objekt der  
Klasse FOLIE



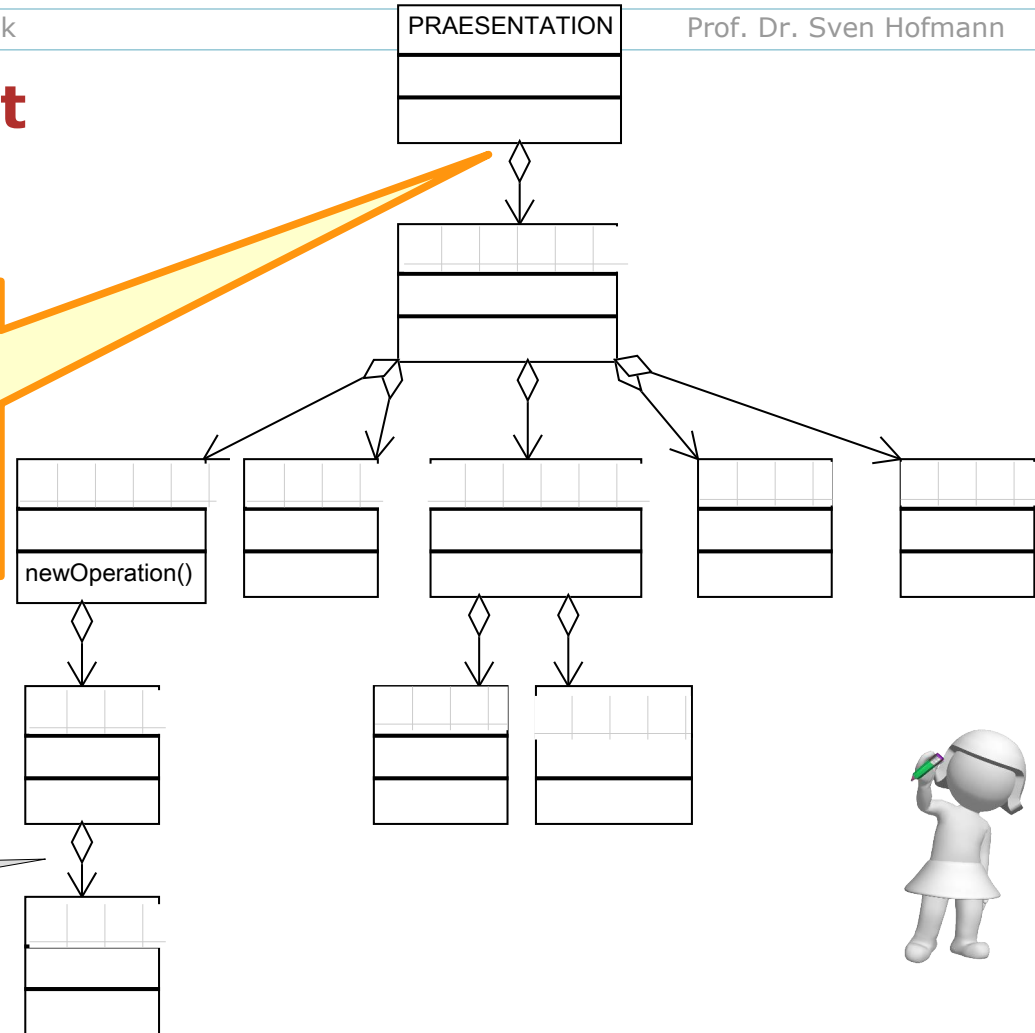
gliederung1: FOLIE
ueberschrift= ..., ..., ... foliennummer= ..., ..., ... folienuebergang= ..., ..., ...
folie_erzeugen() folie_loeschen() folie_kopieren()

# OOM als Softwarekonzept

## Basis-Konzept für Präsentations-Software

Wesentliches Grundprinzip der OOM:  
**Vererbung!**

Objekte einer Sub-Klasse übernehmen („erben“) Attribute und Methoden der Vaterklasse (und ergänzen weitere).



Relation: „ist Teil von“  
 „erbt von“



# OOM als Softwarekonzept

## Basis-Konzept für Präsentations-Software



Überschrift Seite 1

Standard

- Linie...
- Fläche...
- Text...
- Position und Größe...
- Fontwork
- Zeichnen...
- Absatz...
- Anordnung
- Ausrichtung
- Umwandeln
- Beschreibung...
- Name...
- Benutzerdefinierte Animation...
- Interaktion...
- Vorlage bearbeiten...
- Ausschneiden
- Kopieren
- Einfügen

Kontextmenü zur Klasse TEXTFELD

Folie

- Fangpunkt/-linie einfügen...
- Navigator
- Raster
- Fanglinien
- Gliederungsansicht
- Foliensortierung
- Bildschirmpräsentation
- Einfügen

- Seite einrichten...
- Eoliendesign
- Eoliensortierung
- Hintergrundgrafik der Folie festlegen...
- Folie umbenennen
- Hintergrundgrafik des Masters anzeigen
- Objekte des Masters anzeigen
- Neue Folie
- Folie löschen

Mit der rechten Maustaste wird das **Kontextmenü** geöffnet → Anzeige der Attribute der jeweiligen Klasse

Kontextmenü zur Klasse FOLIE

# OOM als Softwarekonzept

## Basis-Konzept für Textverarbeitungs-Software

Das unter dem Dateinamen  
*einladung3.odt* gespeicherte Dokument  
ist ein **Objekt** der Klasse TEXTDOKUMENT



einladung3.odt:TEXTDOKUMENT
autor= Hans Kraftpunkt titel= Interaktiv Lernen datum= 31.12.2000
document_erzeugen() document_oeffnen() document_speichern()

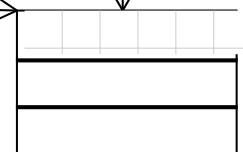
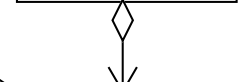
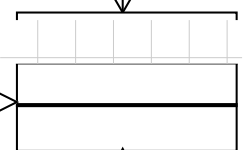
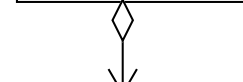
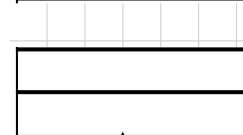
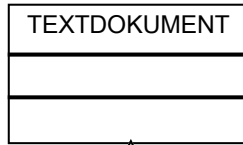
Das Objekt *ueberschrift1*  
ist ein **Objekt** der  
Klasse ABSATZ



ueberschrift1: ABSATZ
ausrichtung= links,block, ... einzug_links= 0cm, ... zeilenabstand= proportional, ...
absatz_erzeugen() absatz_loeschen() absatz_kopieren()

# OOM als Softwarekonzept

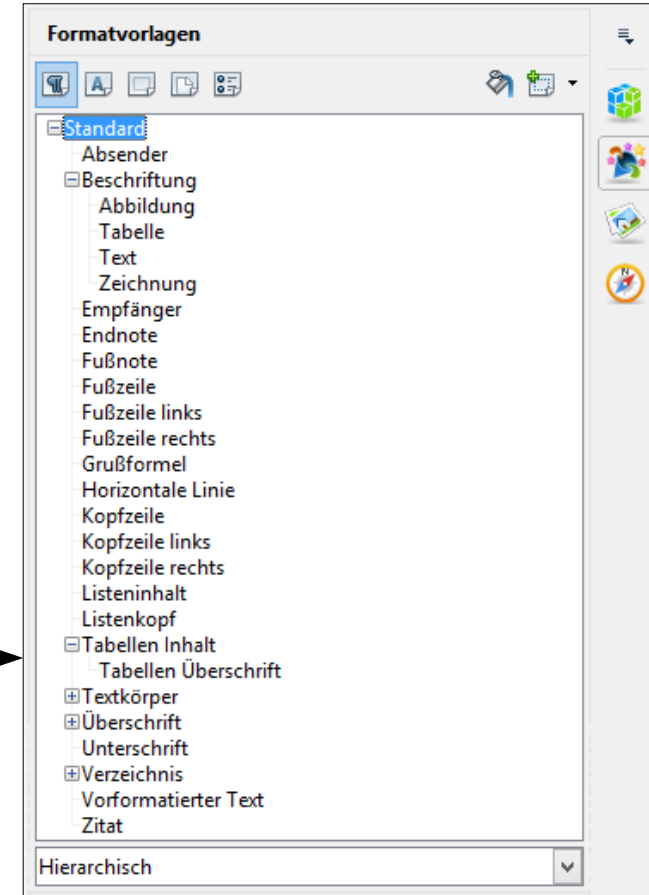
## Basis-Konzept für Textverarbeitungs-Software



**Hinweis:**

**Zur einheitlichen Gestaltung aller Objekte einer Klasse in der Textverarbeitung sollten die Attribut-Werte in den Formatvorlagen eingestellt werden!**

(NICHT für jedes Objekt einzeln!)



# OOM als Softwarekonzept

The screenshot shows the OpenOffice Writer interface with a document titled 'vorlage\_wissArbeit.odt'. The document content includes a title '<Kurztitel>' and 'Kapitel2'. The main text is structured as follows:

- 2 Kapitel2**
- 2.1 Seiten-Layout**
- 2.1.1 Abbildungen Fließtext**

The text under '2.1.1' reads: 'An dieser Stelle wird ein Bild eingefügt, welches eine entsprechende Beschriftung erhält. Am Schluss der Arbeit wird ein automatisch generiertes Inhaltsverzeichnis erstellt. Auf dieses Verzeichnis kann man zugreifen. Dieser Text soll links neben das Bild. Man beachte: Das Korrigieren der Seitenzahl beeinflusst die tatsächliche Seitennummerierung nicht! Zwar wird im Datenfeld die korrigierte Seitenzahl gezeigt, im Inhaltsverzeichnis würde aber die absolute Seitenzahl angezeigt werden. So stünde im Verzeichnis Seite 3, während die erste Textseite als Seite 1 deklariert ist.'

Annotations with hand icons point to various elements:

- 'ueberschrift 2: ABSATZ' points to the '2.1 Seiten-Layout' heading.
- 'seite\_4: SEITE' points to the '2.1.1 Abbildungen Fließtext' heading.
- 'absatz\_21: ABSATZ' points to the paragraph of text.
- 'buchstabe\_T: ZEICHEN' points to the letter 't' in 'tatsächlich'.

A dialog box titled 'Formateigenschaften' is open, showing a list of styles. The 'Seitennummerierung' section is expanded, showing options for 'Absatznummerierung' and 'Seitennummerierung'. The 'Seitennummerierung' section is further expanded, showing options for 'Absatznummerierung' and 'Seitennummerierung'. The 'Seitennummerierung' section is further expanded, showing options for 'Absatznummerierung' and 'Seitennummerierung'. The 'Seitennummerierung' section is further expanded, showing options for 'Absatznummerierung' and 'Seitennummerierung'.

At the bottom of the dialog box, there is a caption: 'Abbildung 1: Korrektur der Seitenzahl über einen Summanden'.

The status bar at the bottom of the window shows 'Seite 2 / 4 / 7', 'Standard', 'Deutsch (Deutschland)', 'EINFG | STD \*', and '135 %'.

# OOM als Softwarekonzept

## Basis-Konzept für Tabellenkalkulations-Software

Das unter dem Dateinamen  
*zensuren\_7a.odc* gespeicherte  
Dokument ist ein Objekt der  
Klasse TABELLENDOKUMENT



zensuren\_7a.odc:TABELLENDOKUMENT

autor= Hans Kraftpunkt  
titel= Zensurenverwaltung\_7a  
datum= 31.12.2000

document\_erzeugen()  
document\_oeffnen()  
document\_speichern()

Das Objekt *klassenliste*  
ist ein Objekt der  
Klasse TABELLE



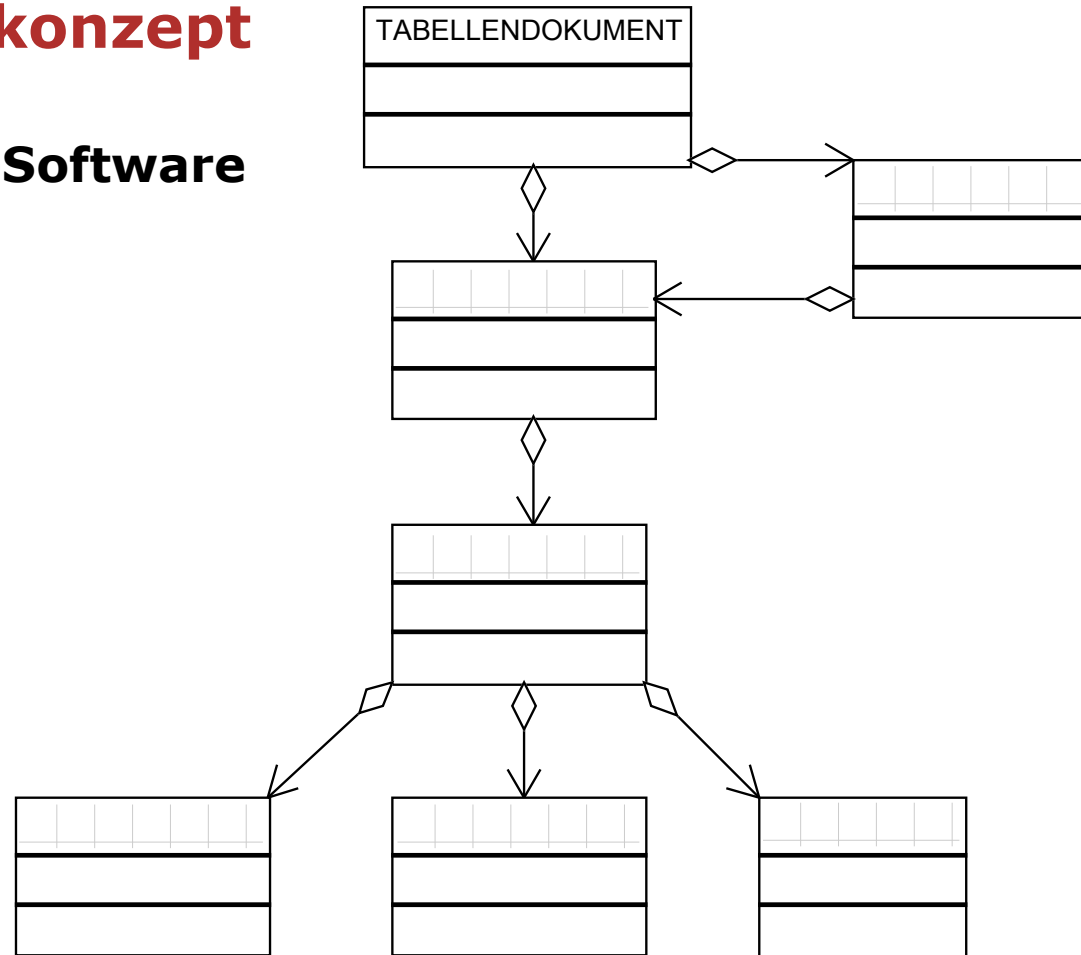
klassenliste: TABELLE

ueberschrift= ...  
druckbereich= ...  
geschützt= ...

tabelle\_erzeugen()  
tabelle\_loeschen()  
tabelle\_kopieren()

# OOM als Softwarekonzept

## Basis-Konzept für Tabellenkalkulations-Software



# OOM als Softwarekonzept

## Basis-Konzept für abellenkalkulations-Software

The screenshot shows an OpenOffice Calc spreadsheet titled 'zensurenverwaltung-klasse-x.ods'. The spreadsheet contains a table of student grades. Annotations with hand icons point to specific elements:

- funktion\_47: FUNKTION**: Points to the formula bar containing `=MITTELWERT(C7:H7)`.
- zelle\_E3: ZELLE**: Points to cell E3, which contains the text 'Fach Deutsch'.
- wert\_3484: WERT**: Points to cell H7, which contains the value '3,4'.
- tabelle\_2: TABELLE**: Points to the 'Tabelle2' tab at the bottom of the spreadsheet.

The spreadsheet data is as follows:

	A	B	C	D	E	F	G	H	I	J	K
1											
2											
3		<b>Fach Deutsch</b>									
4											
5	Name	Vorname	Arbeit 1	Arbeit 2	Arbeit 3	Arbeit 4	Arbeit 5	Arbeit 6	Durchschnitt Arbeit	Sonstige	Sons
6											
7	Arnold	Max	2	5	3	2	5		3,4		6
8	Berthold	Beatrice	4	6	6	6	6		5,6		6
9	Kraftpunkt	Arno		2							
10	Schüler4	Vschüler4			3						

## Literatur:

- Appelrath et al.: Starthilfe Informatik. Teubner, 1998.
- Holzinger, A.: Basiswissen IT/Informatik. Vogel, 2002.
- Jank, W.; Meyer, H.: Didaktische Modelle. Cornelsen. Berlin, 2005.
- Kerner, I.O. et al.: Lehr- und Übungsbuch Informatik (Bd.1 - 4). Fachbuchverlag, 2001.
- Precht, M.: EDV-Grundwissen. Addison Wesley, 1999.
- Erler, T.: Das Einsteigerseminar UML. Bhv-Verlag, 2000.
- Fischer, H.; Knapp, T.: Modellieren im Informatikunterricht der Sekundarstufe I. IN: LOG IN 135 (2005)
- Maier, U.: Lehr-Lernprozesse in der Schule: Studium. Allgemeindidaktische Kategorien für die Analyse und Gestaltung von Unterricht. Julius Klinkhardt, Bad Heilbrunn, 2012.
- Thomas, M.: Informatische Modellbildung. Modellieren von Modellen als ein zentrales Element der Informatik für den allgemeinbildenden Schulunterricht. Potsdam, 2002.

## Internet:

- <https://www.sachsen.schule>, Pädagogische Plattform des Sächsischen Bildungsservers, vom 03.04.2020
- [https://www.sachsen.schule/~fischer/gw\\_neu](https://www.sachsen.schule/~fischer/gw_neu), Grundwissen - neu gemacht, vom 03.04.2020
- <http://www.tinohempel.de/info/mathe/ries/ries.htm>, Adam Ries, vom 03.04.2020
- <https://phet.colorado.edu/de/simulations>

1. Erläutern Sie anhand von Beispielen aus der realen Welt die wesentlichsten Eigenschaften von Modellen. Erörtern Sie unterschiedliche Einsatzzwecke.
2. Grenzen Sie die Begriffe „Unterrichtsform“ und „Unterrichtskonzept“ als Form der Modellierung in der Didaktik voneinander ab.
3. Geben Sie einen Überblick über Modellierungs-Arten in der Informatik!
4. Beschreiben Sie die grundlegenden Gedanken der Objektorientierten Modellierung (OOM)! Erläutern Sie diese anhand geeigneter Beispiele für Klassen und Objekte.
5. Wenden Sie die Grundlagen der OOM auf im Unterricht eingesetzte Software (Präsentation, Textverarbeitung, Tabellenkalkulation) an.  
Leiten Sie daraus Schlussfolgerungen für Ihren Umgang mit dieser Software ab!
6. Ordnen Sie Masterfolien für Präsentationen sowie Formatvorlagen für Textverarbeitung in das Konzept der OOM ein!
7. Erstellen Sie einen Katalog von Hinweisen, die Sie den Schüler-Innen für den Umgang mit Dokumenten (Präsentation, Textverarbeitung, Tabellenkalkulation) zur Verfügung stellen!

