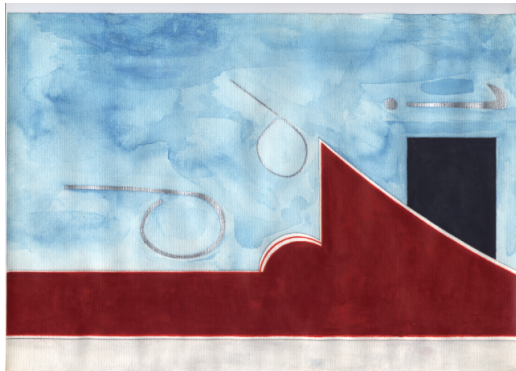


# Didaktik der Informatik – Vorlesung

## 10. Vorlesung: Informatikunterricht – Beispielszenarien

StD Dipl.-Inform. Dr. rer. nat. L. Humbert

Fachgebiet Didaktik der Informatik  
Fachbereich C – Universität Wuppertal



# Gliederung der Präsentation

- 1 Übersicht
  - Gliederung der Präsentation
  - Themen der Vorlesung im Sommersemester 2009
- 2 Informatikunterricht – Beispielszenarien
  - Veranstaltungsziele – Kompetenzen
- 3 Informatische Allgemeinbildung
  - Vision
  - Basis – Kontext
- 4 Szenarien Informatische Allgemeinbildung
  - Kompetenzen der allgemeinbildenden Informatik
- 5 Beispiele – Mittlerer Bildungsabschluss
  - Unterrichtskonzept – Objektorientierte Sicht auf Informatiksysteme
- 6 Szenarien Oberstufe
  - Ziele des Informatikunterrichts in der gymnasialen Oberstufe
- 7 Beispiele – Oberstufe
  - Unterrichtskonzept Stifte & Mäuse (SuM)
  - Beispiellage – SuM
  - Objektorientierte Modellierung (OOM)

# Themen der Vorlesung (Stand 26. Juni 2009)

Kapitel in [Humbert, 2006]

|    |   |          |
|----|---|----------|
| 1  | Organisatorisches – Einführung .....                    | 2, 4     |
| 2  | Informatik – geschichtliche Aspekte .....               | 2        |
| 3  | Genderdiskussion .....                                  | 9        |
| 4  | Grundfragen des Lernens .....                           | 3        |
| 5  | Schulinformatik – Entwicklungslinien .....              | 4        |
| 6  | Schulinformatik – Normierung .....                      | 4        |
| 7  | Informatikunterricht – besondere Arbeitsweisen ..       | 5        |
| 8  | Informatikunterrichtsplanung – Vorgehensmodelle .....   | 6        |
| 9  | Informatikunterrichtsplanung .....                      | 6, 7     |
| 10 | <b>Informatikunterricht – Beispielszenarien .....</b>   | <b>7</b> |
| 11 | Informatikunterricht – Leistungsmessung .....           | 8        |
| 12 | Moralisch-ethische Aspekte – Professionalisierung ..... | 9, 10    |



# Kompetenzen 10. Vorlesung: Informatikunterricht – Beispielszenarien

## Kompetenzen 10. Vorlesung: Informatikunterricht – Beispielszenarien

- Ziele des Informatikunterrichts mit konkreten Beispielen für den Unterricht illustrieren
- Grundlegende Ideen, Konzept und Umsetzung für eine didaktisch gestaltete Klassenbibliothek (Stifte & Mäuse) beispielhaft illustrieren
- Kritische Würdigung und Prüfung der Eignung vornehmen

# Kompetenzen 10. Vorlesung: Informatikunterricht – Beispielszenarien

## Kompetenzen 10. Vorlesung: Informatikunterricht – Beispielszenarien

- Ziele des Informatikunterrichts mit konkreten Beispielen für den Unterricht illustrieren
- Grundlegende Ideen, Konzept und Umsetzung für eine didaktisch gestaltete Klassenbibliothek (Stifte & Mäuse) beispielhaft illustrieren
- Kritische Würdigung und Prüfung der Eignung vornehmen

## [Puhlmann, 2005, S. 79] – Informatische Bildung

Stell' dir eine Schule vor, in der alle Schülerinnen und Schüler hervorragenden Informatikunterricht erhalten. Der Unterricht wird von Lehrerinnen und Lehrern erteilt, die eine fundierte Informatikausbildung haben und die zugleich wissen, wie Informatikinhalte Kindern und Jugendlichen nahe gebracht werden können. Ihnen stehen angemessene Arbeits- und Unterrichtsmittel zur Verfügung, die es erlauben, wichtige Informatikinhalte in methodischer Vielfalt zu behandeln und informatische Kompetenzen bei den Schülerinnen und Schülern zu entwickeln. Dabei sind die Anforderungen durchaus hoch, aber die Schülerinnen und Schüler werden damit nicht alleine gelassen, sondern nach ihren Bedürfnissen unterstützt. So zeigen die Schülerinnen und Schüler großes Engagement, lernen mit Verständnis, erkennen Verbindungen zwischen verschiedenen informatischen Fragestellungen, tauschen sich untereinander über Informatik aus und können Überlegungen und Arbeitsergebnisse mündlich und schriftlich gut verständlich mitteilen. Dabei nutzen sie selbstverständlich Computer sowohl als Gegenstand des Unterrichts als auch als Arbeitsmittel zur Informationsdarstellung und zum Informationsaustausch. Diese Kompetenzen kommen auch ihrer übrigen schulischen Arbeit zugute. So schätzen die Schülerinnen und Schüler das Fach Informatik und engagieren sich stark, um ihr Wissen und ihre Kompetenzen zu

# Basis – Kontext im Rahmen der Vorlesungen

- Informatische Phänomene verstehen (vgl. [Humbert u. Puhlmann, 2005])
- Spiralprinzip (vgl. Vorlesung 4)
- Probleme lösen als zentraler Ausgangs- und Zielpunkt jeden Informatikunterrichts (vgl. Vorlesung 5)
- Projektorientierung als methodische Voraussetzung zur Bearbeitung echter Problemstellungen in der Informatik und im Informatikunterricht (s. v.)
- Modellieren und Konstruieren sowie Analysieren und Bewerten (vgl. Vorlesung 9)

# Basis – Kontext im Rahmen der Vorlesungen

- Informatische Phänomene verstehen (vgl. [Humbert u. Puhlmann, 2005])
- Spiralprinzip (vgl. Vorlesung 4)
- Probleme lösen als zentraler Ausgangs- und Zielpunkt jeden Informatikunterrichts (vgl. Vorlesung 5)
- Projektorientierung als methodische Voraussetzung zur Bearbeitung echter Problemstellungen in der Informatik und im Informatikunterricht (s. v.)
- Modellieren und Konstruieren *sowie* Analysieren und Bewerten (vgl. Vorlesung 9)

# Basis – Kontext im Rahmen der Vorlesungen

- Informatische Phänomene verstehen (vgl. [Humbert u. Puhlmann, 2005])
- Spiralprinzip (vgl. Vorlesung 4)
- Probleme lösen als zentraler Ausgangs- und Zielpunkt jeden Informatikunterrichts (vgl. Vorlesung 5)
- Projektorientierung als methodische Voraussetzung zur Bearbeitung echter Problemstellungen in der Informatik und im Informatikunterricht (s. v.)
- Modellieren und Konstruieren *sowie* Analysieren und Bewerten (vgl. Vorlesung 9)

# Basis – Kontext im Rahmen der Vorlesungen

- Informatische Phänomene verstehen (vgl. [Humbert u. Puhlmann, 2005])
- Spiralprinzip (vgl. Vorlesung 4)
- Probleme lösen als zentraler Ausgangs- und Zielpunkt jeden Informatikunterrichts (vgl. Vorlesung 5)
- Projektorientierung als methodische Voraussetzung zur Bearbeitung echter Problemstellungen in der Informatik und im Informatikunterricht (s. v.)
- Modellieren und Konstruieren *sowie* Analysieren und Bewerten (vgl. Vorlesung 9)

# Basis – Kontext im Rahmen der Vorlesungen

- Informatische Phänomene verstehen (vgl. [Humbert u. Puhlmann, 2005])
- Spiralprinzip (vgl. Vorlesung 4)
- Probleme lösen als zentraler Ausgangs- und Zielpunkt jeden Informatikunterrichts (vgl. Vorlesung 5)
- Projektorientierung als methodische Voraussetzung zur Bearbeitung echter Problemstellungen in der Informatik und im Informatikunterricht (s. v.)
- Modellieren und Konstruieren *sowie* Analysieren und Bewerten (vgl. Vorlesung 9)

# Basis – Kontext im Rahmen der Vorlesungen

- Informatische Phänomene verstehen (vgl. [Humbert u. Puhlmann, 2005])
- Spiralprinzip (vgl. Vorlesung 4)
- Probleme lösen als zentraler Ausgangs- und Zielpunkt jeden Informatikunterrichts (vgl. Vorlesung 5)
- Projektorientierung als methodische Voraussetzung zur Bearbeitung echter Problemstellungen in der Informatik und im Informatikunterricht (s. v.)
- Modellieren und Konstruieren *sowie* Analysieren und Bewerten (vgl. Vorlesung 9)

# Ziele Mittlerer Bildungsabschluss

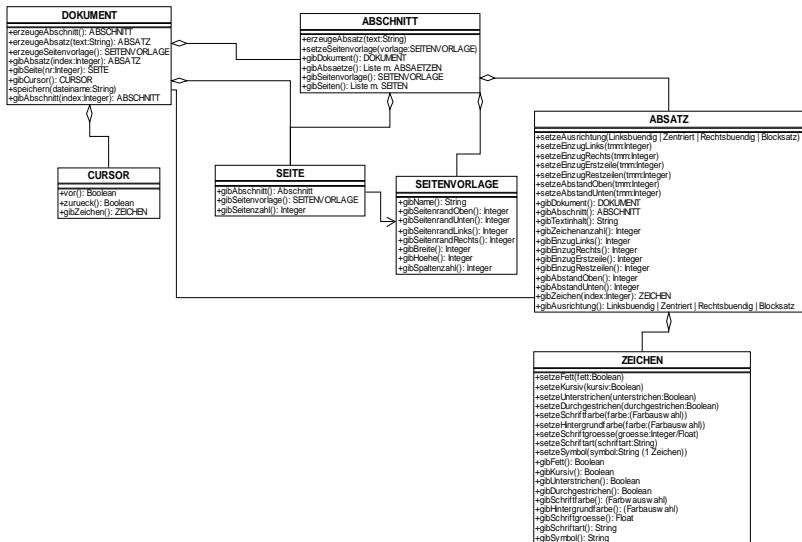
## **Informatische Bildung als Teil allgemeiner Bildung**

Das übergeordnete **Ziel informatischer Bildung** in Schulen ist es, Schülerinnen und Schüler bestmöglich auf ein Leben in einer Informationsgesellschaft vorzubereiten [...]

**Jede Schülerin und jeder Schüler** soll dazu in die Lage versetzt werden, auf einem der jeweiligen Schulart angemessenen Niveau den grundlegenden Aufbau von »Informatiksystemen« und deren Funktionsweise zu verstehen [...]











Die schulische Auseinandersetzung mit dem Aufbau und der Funktionsweise von Informatiksystemen darf dabei aber **nicht** nur auf der Ebene der Benutzungsschnittstelle erfolgen, die sich bereits bei einer nächsten Produktversion oder bei Verwendung eines Produkts eines anderen Herstellers ändern kann. Den Ausgangspunkt für einen produktunabhängigen Zugang bildet daher die »Darstellung« bzw. Repräsentation von »Information« zu Problemen aus der Lebenswelt der Schülerinnen und Schüler durch »Daten« in Informatiksystemen verschiedener Anwendungsklassen  
[GI, 2008, S. 11 (Hervorhebungen durch lh)].

# Klassendiagramm – Ponto



# Dokumente – objektorientiert (1/3)

nach [Borchel u. a., 2005]

| DOKUMENT  |  |
|---|--|
|  | Name: ZEICHENKETTE                       |
|  | Abschnitte: LISTE – ABSCHNITT            |
|  | Speicherort: ZEICHENKETTE                |
|  | erzeugeAbsatz(text:ZEICHENKETTE): ABSATZ |
|  | gibAbsatz(index:GANZEZAHL): ABSATZ       |
|  | erzeugeAbschnitt(): ABSCHNITT            |
|  | gibCursor(): CURSOR                      |
|  | lade(ort:ZEICHENKETTE): WAHRHEITSWERT    |
|  | speichere(ort:ZEICHENKETTE)              |
|  | schlieÙe()                               |

## Dokumente – objektorientiert (2/3)

### absatz1

Zeichen = "Hallo, liebe Freundinnen und Freunde,"

Ausrichtung = Zentriert

Zeilenabstand = 1,5

EinzugLinks= 2000

EinzugRechts = 2000

EinzugErstzeile = 0

EinzugRestzeilen = 0

AbstandOben = 0

AbstandUnten = 0

## Dokumente – objektorientiert (3/3)

Umsetzung der Modellierung in einer Programmiersprache – hier mit Python und der Schnittstelle Ponto zu OpenOffice.org

```
1 from _ponto_ import _DOKUMENT, Zentriert
2 einladung=_DOKUMENT()
3 absatz1=einladung.erzeugeAbsatz(" Hallo , _liebe _Freundinnen _und _Freunde , ")
4 absatz1.setzeAusrichtung(Zentriert)
5 ...
```

**Ponto** ist öffentlich zugänglich – vgl. [Reinertz, 2008]. Unter [http://themartin.110mb.com/moodle\\_classes/mod/resource/view.php?id=18](http://themartin.110mb.com/moodle_classes/mod/resource/view.php?id=18) finden sich der Quellcode (ponto.py) und ablauffähige Beispiele

# Ziele gymnasiale Oberstufe

## Allgemeine Bildung plus Propädeutik im Fach

- Input: [noch] gültige Richtlinien und Lehrplan – öffentlich verfügbar [MSWWF, 1999]  
der Lehrplan Informatik befindet sich – wie die Lehrpläne der anderen Fächer auch – in Überarbeitung; da er Elemente enthält, die bezüglich des Zentralabiturs als nicht mehr zielführend betrachtet werden [müssen]
- Output: Zentralabitur – **Aufgaben** (2007 und 2008) Aufgabenbeispiele und Rahmenvorgaben (zur Zeit für 2009, 2010, 2011) **öffentlich** verfügbar

### Abitur 2007–2011

<http://www.standardsicherung.schulministerium.nrw.de/abitur-gost/fach.php?fach=15>  
geprüft: 6. Juli 2009

# Das Konzept Stifte & Mäuse



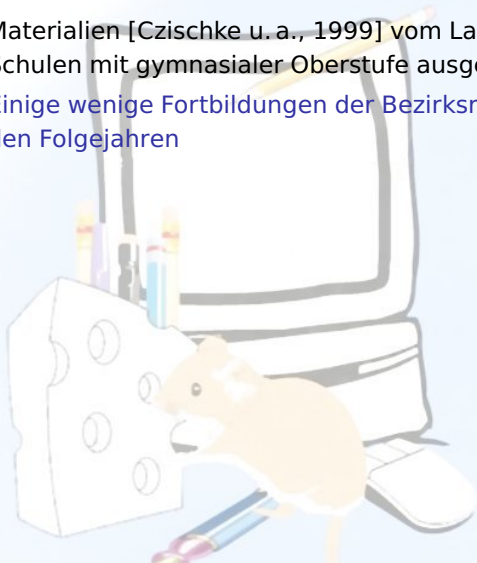
# Das Konzept Stifte & Mäuse

- Materialien [Czischke u. a., 1999] vom Landesinstitut Soest 1999 an alle Schulen mit gymnasialer Oberstufe ausgeliefert



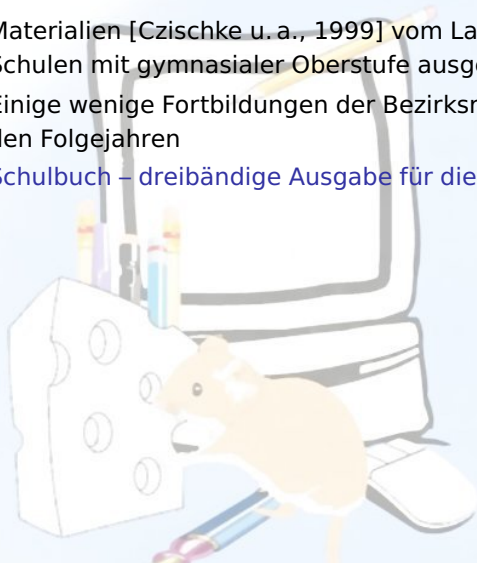
# Das Konzept Stifte & Mäuse

- Materialien [Czischke u. a., 1999] vom Landesinstitut Soest 1999 an alle Schulen mit gymnasialer Oberstufe ausgeliefert
- Einige wenige Fortbildungen der Bezirksregierungen für Kolleginnen in den Folgejahren



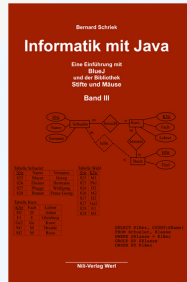
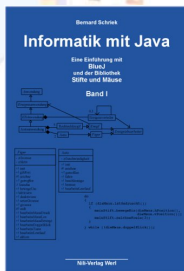
# Das Konzept Stifte & Mäuse

- Materialien [Czischke u. a., 1999] vom Landesinstitut Soest 1999 an alle Schulen mit gymnasialer Oberstufe ausgeliefert
- Einige wenige Fortbildungen der Bezirksregierungen für Kolleginnen in den Folgejahren
- Schulbuch – dreibändige Ausgabe für die Jahrgänge 11, 12 und 13



# Das Konzept Stifte & Mäuse

- Materialien [Czischke u. a., 1999] vom Landesinstitut Soest 1999 an alle Schulen mit gymnasialer Oberstufe ausgeliefert
- Einige wenige Fortbildungen der Bezirksregierungen für Kolleginnen in den Folgejahren
- Schulbuch – dreibändige Ausgabe für die Jahrgänge 11, 12 und 13
  - verfügbar: [Schriek, 2005, 2006, 2007]



# Details – SuM

- Klassenbibliothek – Anforderungen (Ende der 90ger)

# Details – SuM

- Klassenbibliothek – Anforderungen (Ende der 90er)
  - Didaktisch gestaltet
    - »Make it as simple as possible, but not simpler« (Einstein)

# Details – SuM

- Klassenbibliothek – Anforderungen (Ende der 90er)
  - Didaktisch gestaltet
    - »Make it as simple as possible, but not simpler« (Einstein)
  - Programmiersprachenunabhängigkeit

# Details – SuM

- Klassenbibliothek – Anforderungen (Ende der 90er)
  - Didaktisch gestaltet
    - »Make it as simple as possible, but not simpler« (Einstein)
  - Programmiersprachenunabhängigkeit
  - Betriebssystemunabhängigkeit

# Details – SuM

- Klassenbibliothek – Anforderungen (Ende der 90er)
  - Didaktisch gestaltet
    - »Make it as simple as possible, but not simpler« (Einstein)
  - Programmiersprachenunabhängigkeit
  - Betriebssystemunabhängigkeit

## Klassen Einstiegsphase

- Bildschirm
- Tastatur
- Maus
- Stift
- Buntstift
- Sprite (Python)

# Details – SuM

- Klassenbibliothek – Anforderungen (Ende der 90er)
  - Didaktisch gestaltet
    - »Make it as simple as possible, but not simpler« (Einstein)
  - Programmiersprachenunabhängigkeit
  - Betriebssystemunabhängigkeit

## Klassen Einstiegsphase

- Bildschirm
- Tastatur
- Maus
- Stift
- Buntstift
- Sprite (Python)

## Implementierung (zeitliche Abfolge)

- Oberon[-2]
- Object Pascal (Macintosh)
- Object Pascal (Window)
- .....einige Jahre später .....
- Java, Python

# Beispiele – SuM

**Problem** Lehrperson, die die Objektorientierte Modellierung (noch) nicht kennt

**Lösung** [sehr | zu?] einfach[st]e Beispiele, die aufbauend nach und nach Schreibweise(n) und Konzepte einführen

**Synthese** Lehrpersonen vernünftig qualifizieren

**Kritik** Modellierung trivial, Modellierung fachlich (mindestens) diskussionswürdig

vgl. <http://oszhdl.be.schule.de/gymnasium/faecher/informatik/didaktik/sum/sum-kritik.htm>

## Beispiel aus SuM

- Original [Czischke u. a., 1999, S. 49] – Pseudosprache
- Humbert – Struktogrammdarstellung
- Schriek [Schriek, 2005, S. 25, 33, 37] – Java
- Linkweiler [Linkweiler, 2002] – Python

# Beispiele – SuM

**Problem** Lehrperson, die die Objektorientierte Modellierung (noch) nicht kennt

**Lösung** [sehr | zu?] einfach[st]e Beispiele, die aufbauend nach und nach Schreibweise(n) und Konzepte einführen

**Synthese** Lehrpersonen vernünftig qualifizieren

**Kritik** Modellierung trivial, Modellierung fachlich (mindestens) diskussionswürdig

vgl. <http://oszhdl.be.schule.de/gymnasium/faecher/informatik/didaktik/sum/sum-kritik.htm>

## Beispiel aus SuM

- Original [Czischke u. a., 1999, S. 49] – Pseudosprache
- Humbert – Struktogrammdarstellung
- Schriek [Schriek, 2005, S. 25, 33, 37] – Java
- Linkweiler [Linkweiler, 2002] – Python

# Pseudosprache

Beispiel nach [Czischke u. a., 1999, S. 49]

Kontext: Kontrollstrukturen – Freihandzeichnen und Radieren

```
1 Stift.hebe dich
2 Wiederhole
3   Wenn Maus.meldet Mausknopf ist gedrückt, dann
4     Stift.bewege dich zur Mausposition
5     Stift.senke dich
6 sonst
7   Stift.hebe dich
8   Wenn Tastatur.meldet Taste wurde gedrückt, dann
9     Stift.wähle Radiermodus
10 bis Maus.meldet Mausknopf wird doppelt geklickt
```

Unschön: die Bezeichner der verwendeten Objekte haben einen großen Anfangsbuchstaben – dies ist verwirrend, da üblicherweise Objekte mit einem kleinen Buchstaben beginnen, aber für Klassen als erstes ein Großbuchstabe verwendet wird.

# Pseudosprache

Beispiel nach [Czischke u. a., 1999, S. 49]

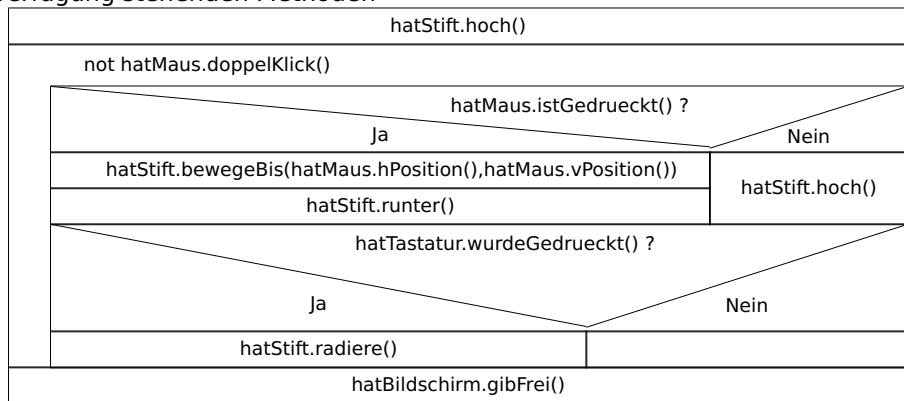
Kontext: Kontrollstrukturen – Freihandzeichnen und Radieren

```
1 Stift.hebe dich
2 Wiederhole
3   Wenn Maus.meldet Mausknopf ist gedrückt, dann
4     Stift.bewege dich zur Mausposition
5     Stift.senke dich
6 sonst
7   Stift.hebe dich
8   Wenn Tastatur.meldet Taste wurde gedrückt, dann
9     Stift.wähle Radiermodus
10 bis Maus.meldet Mausknopf wird doppelt geklickt
```

Unschön: die Bezeichner der verwendeten Objekte haben einen großen Anfangsbuchstaben – dies ist verwirrend, da üblicherweise Objekte mit einem kleinen Buchstaben beginnen, aber für Klassen als erstes ein Großbuchstabe verwendet wird.

# Struktogramm

Umsetzung der Pseudosprache mit den durch die Klassenbibliothek zur Verfügung stehenden Methoden



Die Objektbezeichner dokumentieren, dass hier eine Objektstruktur mit hat-Beziehungen (Kompositionen) nahegelegt wird

# Java – (1/2)

```
1 import _sum.kern.Bildschirm ;
2 import _sum.kern.Stift ;
3 import _sum.kern.Maus ;
4 import _sum.kern.Tastatur ;
5
6 public _class _StartFreihand5
7 {
8     _ _Bildschirm _hatBildschirm ;
9     _ _Stift _hatStift ;
10    _ _Maus _hatMaus ;
11    _ _Tastatur _hatTastatur ;
12
13    _public _StartFreihand5 () ;
14    _ _ {
15        _ _ _hatBildschirm _=_new _Bildschirm () ;
16        _ _ _hatStift _=_new _Stift () ;
17        _ _ _hatMaus _=_new _Maus () ;
18        _ _ _hatTastatur _=_new _Tastatur () ;
19    }
}
```

# Java – (2/2)

```
21 public void fuehreAus();
22 {
23     hatStift.hoch();
24     do
25     {
26         if (hatMaus.istGedueckt())
27         {
28             hatStift.bewegeBis(hatMaus.hPosition(), hatMaus.vPosition());
29             hatStift.runter();
30         }
31     } else
32     {
33         hatStift.hoch();
34     }
35     if (hatTastatur.wurdeGedueckt())
36     {
37         hatStift.radiere();
38     }
39 } while (!hatMaus.doppelKlick());
40 hatBildschirm.gibFrei();
41 }
42
43 }
```

# Python

```
1 from _sum_ import _Bildschirm , Stift , Maus, Tastatur
2 hatBildschirm=_Bildschirm()
3 hatStift=_Stift(hatBildschirm)
4 hatMaus=_Maus(hatBildschirm)
5 hatTastatur=_Tastatur(hatBildschirm)
6
7 hatStift.hoch()
8 while _not_ hatMaus.doppelKlick():
9     if _hatMaus.istGedruickt():
10         hatStift.bewegeBis(hatMaus.hPosition(),
11                             hatMaus.vPosition())
12         hatStift.runter()
13     else:
14         hatStift.hoch()
15     if _hatTastatur.wurdeGedruickt():
16         hatStift.radiere()
17
18 hatBildschirm.gibFrei()
```

# OOM unter Benutzung der Klassenbibliothek

- 1 Zeichnen – »Malen mit der Maus«

# OOM unter Benutzung der Klassenbibliothek

- 1 Zeichnen – »Malen mit der Maus«
- 2 Login-Vorgang als Einstieg [Eickhoff u. a., 2005]  
benötigt zusätzlich die Klasse Rahmen

# OOM unter Benutzung der Klassenbibliothek

- 1 Zeichnen – »Malen mit der Maus«
- 2 Login-Vorgang als Einstieg [Eickhoff u. a., 2005]  
benötigt zusätzlich die Klasse Rahmen
- 3 Bücherkauf [Humbert 2006a] unveröffentlicht

# OOM unter Benutzung der Klassenbibliothek

- 1 Zeichnen – »Malen mit der Maus«
- 2 Login-Vorgang als Einstieg [Eickhoff u. a., 2005]  
benötigt zusätzlich die Klasse Rahmen
- 3 Bücherkauf [Humbert 2006a] unveröffentlicht
- 4 ... dieser Zug hat Verspätung [Boettcher u. a., 2007]

# OOM unter Benutzung der Klassenbibliothek

- 1 Zeichnen – »Malen mit der Maus«
- 2 Login-Vorgang als Einstieg [Eickhoff u. a., 2005]  
benötigt zusätzlich die Klasse Rahmen
- 3 Bücherkauf [Humbert 2006a] unveröffentlicht
- 4 ... dieser Zug hat Verspätung [Boettcher u. a., 2007]

# OOM unter Benutzung der Klassenbibliothek

- ① Zeichnen – »Malen mit der Maus«
  - ② Login-Vorgang als Einstieg [Eickhoff u. a., 2005]  
benötigt zusätzlich die Klasse Rahmen
  - ③ Bücherkauf [Humbert 2006a] unveröffentlicht
  - ④ ... dieser Zug hat Verspätung [Boettcher u. a., 2007]
- ① ist – gemessen an didaktischen Kriterien, wie Lebensweltbezug usw. – in einen größeren Kontext einzubinden (z. B. CAD-System)

# OOM unter Benutzung der Klassenbibliothek

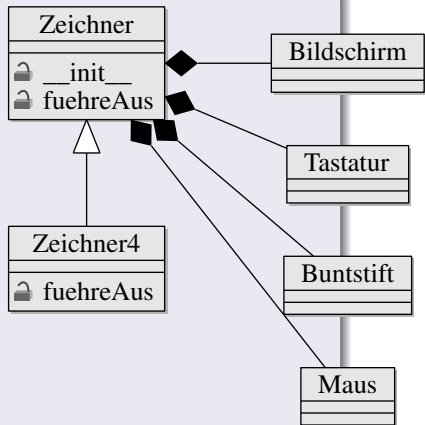
- ① Zeichnen – »Malen mit der Maus«
- ② Login-Vorgang als Einstieg [Eickhoff u. a., 2005]  
benötigt zusätzlich die Klasse Rahmen
- ③ Bücherkauf [Humbert 2006a] unveröffentlicht
- ④ ... dieser Zug hat Verspätung [Boettcher u. a., 2007]

① ist – gemessen an didaktischen Kriterien, wie Lebensweltbezug usw. – in einen größeren Kontext einzubinden (z. B. CAD-System)

② ... ④ gehen von einer realweltlichen Erfahrung aus – werden als Modellierungsaufgaben betrachtet, bearbeitet und führen nicht zu der Klassenstruktur, wie sie durch das Konzept SuM vorgegeben wird, sondern zu einer fachlich zu modellierenden Problemsicht. Im Zusammenhang mit konkreten Darstellungen wird die Klassenstruktur aus dem Konzept SuM benutzt. Die Klassenbibliothek wird primär als Hilfsmittel für grafische Darstellungen genutzt.

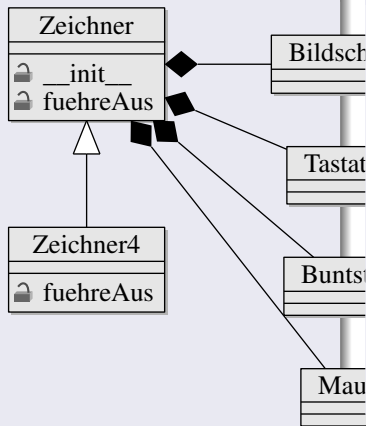
# OOM – Modellierung von ① mit den Klassen aus SuM

## Klassenstruktur



# OOM – Modellierung von ① mit den Klassen aus SuM

## Klassenstruktur



## Klassenbeziehungen

- Zeichner4 (**Unterklasse**) **erbt** von Zeichner (**Oberklasse**)
  - Pfeil gemäß UML-Notation: offenes Dreieck – zeigt auf die Oberklasse
- Zeichner (**Ganzes**) **hat** Bildschirm, Maus usw.  
diese Klassenbeziehung wird fachlich als **Komposition** bezeichnet
  - existenzabhängig, d. h. Zeichner ist für die Erzeugung und den Abbau verantwortlich
  - die gefüllte Raute zeigt auf das **Ganze**

# OOM – Implementierung von ① mit SuM-Klassen

```
1 from Freihand import Zeichner
2
3 class Zeichner4(Zeichner):
4
5     def fuehreAus(self):
6         while not self.hatMaus.doppelKlick():
7             self.hatBuntstift.hoch()
8             if self.hatMaus.istGedruickt():
9                 self.hatBuntstift.runter()
10                self.hatBuntstift.bewegeBis(self.hatMaus.hPosition(),
11                                             self.hatMaus.vPosition())
12            self.derBildschirm.gibFrei()
13
14 if __name__ == "__main__":
15     derZeichner= Zeichner4()
16     derZeichner.fuehreAus()
```

**Problem:** bei den meisten der vorgeschlagenen Beispiele wird immer **genau ein** Objekt jeder Klasse instanziiert – dies widerspricht aktuellen Studien zum Lernen objektorientierter Konzepte

# OOM – Alternative – ③ zunächst »richtige Modellierung«

## Problemstellung

Sie befinden sich bei der Buchhändlerin Ihres Vertrauens und möchten zwei Bücher erwerben: eines der Bücher soll sich mit einem Thema beschäftigen, das mit Informatik zusammenhängt, das zweite soll der Entspannung dienen.

# OOM – Alternative – ③ zunächst »richtige Modellierung«

## Problemstellung

Sie befinden sich bei der Buchhändlerin Ihres Vertrauens und möchten zwei Bücher erwerben: eines der Bücher soll sich mit einem Thema beschäftigen, das mit Informatik zusammenhängt, das zweite soll der Entspannung dienen.

## Arbeitsauftrag

Geben Sie die Objekte an, die in der Problemstellung auftreten. Identifizieren Sie Attribute, die die Objekte für die Lösung der Aufgabe kennzeichnen. Überlegen Sie, über welche Methoden die Objekte »von sich aus« verfügen müssen, damit das Problem einer Lösung zugeführt werden kann.

# OOM – Alternative – ③ zunächst »richtige Modellierung«

## **Objekte – Attribute (und Attributwerte) – Methode(n)**

buchhaendlerin – Ort= BUCHHANDLUNG – verkaufeBuch( buch )

informatikBuch – Thema=INFORMATIK – binGelesen()

entspannungsbuch – Thema=ENTSPANNUNG – binGelesen()

kunde – Ort= BUCHHANDLUNG – erwirbBuch( buch )

# Modellierung – ③ Objektkarten

## **buchhaendlerin**

Ort= BUCHHANDLUNG

...

verkaufeBuch( buch )

## **entspannungsbuch**

Thema= ENTSPANNUNG

...

binGelesen()

## **informatikbuch**

Thema= INFORMATIK

...

binGelesen()

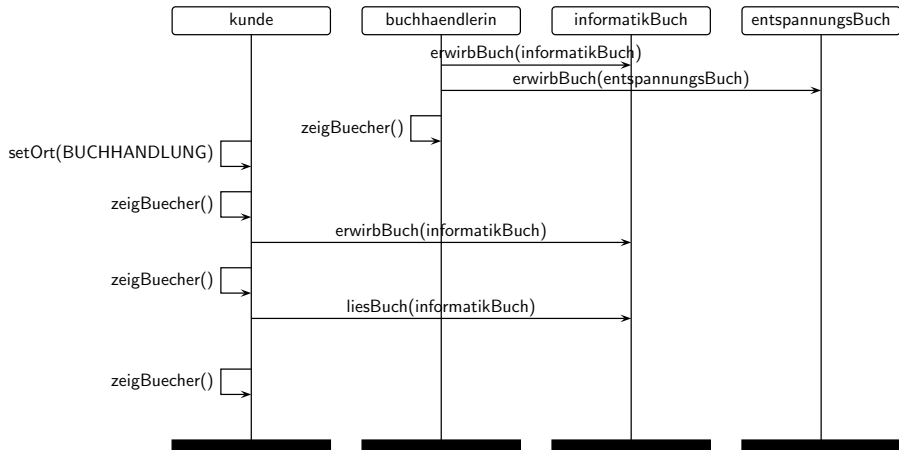
## **kunde**

Ort= BUCHHANDLUNG

Vertrauen= ...

erwirbBuch( buch )

# Modellierung – ③ Sequenzdiagramm



# Modellierung – ③ Quellcode

## Quellcode – Sequenz

```
buchhaendlerin= Person (BUCHHANDLUNG)
informatikBuch= Buch (INFORMATIK)
entspannungsbuch= Buch (ENTSPANNUNG)

buchhaendlerin.erwirbBuch(informatikBuch)
buchhaendlerin.erwirbBuch(entspannungsbuch)

print 'Bücher der Buchhändlerin:'
buchhaendlerin.zeigeBuecher()

kunde= Person ('in der Schule')
kunde.setOrt(BUCHHANDLUNG)

print 'Bücher des Kunden:'
kunde.zeigeBuecher()
kunde.erwirbBuch(entspannungsbuch)
buchhaendlerin.verkaufeBuch(entspannungsbuch)

print 'Bücher des Kunden:'
kunde.zeigeBuecher()
kunde.liesBuch(informatikBuch)

print 'Kunde erwirbt Informatikbuch:'
kunde.erwirbBuch(informatikBuch)
buchhaendlerin.verkaufeBuch(informatikBuch)

print 'Bücher des Kunden:'
kunde.zeigeBuecher()

print 'Kunde liest Informatikbuch'
kunde.liesBuch(informatikBuch)
kunde.zeigeBuecher()
```

# Modellierung – ③ Quellcode

## Quellcode – Sequenz

```

buchhaendlerin= Person (BUCHHANDLUNG)
informatikBuch= Buch (INFORMATIK)
entspannungsbuch= Buch (ENTSPANNUNG)

buchhaendlerin.erwirbBuch(informatikBuch)
buchhaendlerin.erwirbBuch(entspannungsbuch)

print 'Bücher der Buchhändlerin:'
buchhaendlerin.zeigBuecher()

kunde= Person ('in der Schule')
kunde.setOrt(BUCHHANDLUNG)

print 'Bücher des Kunden:'
kunde.zeigBuecher()
kunde.erwirbBuch(entspannungsbuch)
buchhaendlerin.verkaufeBuch(entspannungsbuch)

print 'Bücher des Kunden:'
kunde.zeigBuecher()
kunde.liesBuch(informatikBuch)

print 'Kunde erwirbt Informatikbuch:'
kunde.erwirbBuch(informatikBuch)
buchhaendlerin.verkaufeBuch(informatikBuch)

print 'Bücher des Kunden:'
kunde.zeigBuecher()

print 'Kunde liest Informatikbuch'
kunde.liesBuch(informatikBuch)
kunde.zeigBuecher()

```

## Hinweise

Die Erstellung der Klassen **Person** und **Buch** wird in der Vorlesung nicht thematisiert – siehe Vorlesung zu C++  
 Im Unterricht kann in dieser Phase des Anfangsunterrichts eine Implementierung dieser Klassen durch die Lehrkraft vorgegeben werden.

# Literatur

- [Boettcher u. a. 2007] Boettcher, Daniel ; Grabowsky, Astrid ; Humbert, Ludger ; Poth, Oliver ; Pumplün, Constanze ; Schulte, Jörg: Ada – dieser Zug hat Verspätung. In: Schubert, Sigrid (Hrsg.): *Informatik und Schule – Didaktik der Informatik in Theorie und Praxis – INFOS 2007 – 12. GI-Fachtagung 19.–21. September 2007, Siegen*. Bonn : Gesellschaft für Informatik, Köllen Druck + Verlag GmbH, September 2007 (GI-Edition – Lecture Notes in Informatics – Proceedings P 112). – ISBN 978-3-88579-206-2, S. 217-228. – <http://www.ham.nw.schule.de/pub/bscw.cgi/1001803> – geprüft: 20. Januar 2008
- [Borchel u. a. 2005] Borchel, Christiane ; Humbert, Ludger ; Reinertz, Martin: Design of an Informatics System to Bridge the Gap Between Using and Understanding in Informatics. In: Micheuz, Peter (Hrsg.) ; Antonitsch, Peter (Hrsg.) ; Mittermeir, Roland (Hrsg.): *Innovative Concepts for Teaching Informatics. Informatics in Secondary Schools: Evolution and Perspectives – Klagenfurt, 30<sup>th</sup> March to 1<sup>st</sup> April 2005*. Wien : Ueberreuter Verlag, 2005. – ISBN 3-8000-5167-2, S. 53-63
- [Czischke u. a. 1999] Czischke, Jürgen ; Dick, Georg ; Hildebrecht, Horst ; Humbert, Ludger ; Ueding, Werner ; Wallos, Klaus ; Landesinstitut für Schule und Weiterbildung (Hrsg.): *Von Stiften und Mäusen*. 1. Aufl. Bönen : DruckVerlag Kettler GmbH, 1999. – ISBN 3-8165-4165-8
- [Eickhoff u. a. 2005] Eickhoff, Patrick ; Figgen, Bernd ; Hammersen, Thomas ; Humbert, Ludger ; Pommerenke, Dirk ; Richter, Detlef ; Striewe, Jörg: Informatik – innovative Konzepte zur Gestaltung einer offenen Anfangssequenz mit vielfältigen Erweiterungen. In: **[Friedrich, 2005]**, S. 263-274. – [http://www.ham.nw.schule.de/pub/bscw.cgi/d166569/INF0S05\\_S-263-274.pdf](http://www.ham.nw.schule.de/pub/bscw.cgi/d166569/INF0S05_S-263-274.pdf) – geprüft: 3. September 2007

# Literatur (cont.)

- [Friedrich 2005] Friedrich, Steffen (Hrsg.): *Informatik und Schule – Informatikunterricht – Konzepte und Realisierung – INFOS 2005 – 11. GI-Fachtagung 28.–30. September 2005, Dresden*. Bonn : Gesellschaft für Informatik, Köllen Druck + Verlag GmbH, September 2005 (GI-Edition – Lecture Notes in Informatics – Proceedings P 60). – ISBN 3–88579–389–X
- [GI 2008] GI: *Grundsätze und Standards für die Informatik in der Schule – Bildungsstandards Informatik für die Sekundarstufe I*. April 2008. – Beschluss des GI-Präsidiums vom 24. Januar 2008 – veröffentlicht als Beilage zur LOG IN 28 (2008) Heft 150/151 [http://www.gi-ev.de/fileadmin/gliederungen/fb-iad/fa-ibs/Empfehlungen/bildungsstandards\\_2008.pdf](http://www.gi-ev.de/fileadmin/gliederungen/fb-iad/fa-ibs/Empfehlungen/bildungsstandards_2008.pdf) – geprüft: 18. Januar 2009. ISSN 0720–8642
- [Humbert 2006] Humbert, Ludger: *Didaktik der Informatik – mit praxiserprobtem Unterrichtsmaterial*. 2., überarbeitete und erweiterte Aufl. Wiesbaden : B.G. Teubner Verlag, 2006 (Leitfäden der Informatik). – ISBN 3–8351–0112–9. – <http://humbert.in.hagen.de/ddi/> – geprüft: 8. März 2009
- [Humbert u. Puhlmann 2005] Humbert, Ludger ; Puhlmann, Hermann: Essential Ingredients of Literacy in Informatics. In: *8<sup>th</sup> IFIP World Conference on Computers in Education, 4–7<sup>th</sup> July 2005, University of Stellenbosch*. Cape Town, South Africa : Document Transformation Technologies cc, July 2005. – ISBN 1–920–01711–9. – Documents/445.pdf
- [Linkweiler 2002] Linkweiler, Ingo: *Eignet sich die Skriptsprache Python für schnelle Entwicklungen im Softwareentwicklungsprozess? – Eine Untersuchung der Programmiersprache Python im softwaretechnischen und fachdidaktischen Kontext*. Dortmund, Universität, Fachbereich Informatik, Fachgebiet Didaktik der Informatik, Diplomarbeit, November 2002. – <http://www.ingo-linkweiler.de/diplom/Diplomarbeit.pdf> – geprüft: 5. Juni 2009

# Literatur (cont.)

- [MSWWF 1999] MSWWF (Hrsg.): *Richtlinien und Lehrpläne für die Sekundarstufe II – Gymnasium/Gesamtschule in Nordrhein-Westfalen – Informatik*. 1. Aufl. Frechen : Ritterbach Verlag, 1999 (Schriftenreihe Schule in NRW 4725). – MSWWF (Ministerium für Schule und Weiterbildung, Wissenschaft und Forschung des Landes Nordrhein-Westfalen)
- [Puhlmann 2005] Puhlmann, Hermann: Bildungsstandards Informatik – zwischen Vision und Leistungstests. In: **[Friedrich, 2005]**, S. 79–89
- [Reinertz 2008] Reinertz, Martin: *Ponto – Objektorientierung mit Openoffice.org Writer. Vorträge, Materialien, Beispiele (inkl. Quellcode: ponto.py)*. März 2008. –  
[http://themartin.110mb.com/moodle\\_classes/course/view.php?id=4](http://themartin.110mb.com/moodle_classes/course/view.php?id=4) – geprüft: 27. Februar 2009
- [Schriek 2005] Schriek, Bernard: *Informatik mit Java. Eine Einführung mit BlueJ und der Bibliothek Stifte und Mäuse*. Werl : Nili-Verlag, 2005 ( Band I). – ISBN 3–00–017092–8. – Kapitel 1–6 (von 13):  
<http://www.mg-werl.de/sum/00P-Buch1.pdf> – geprüft: 17. Juni 2008
- [Schriek 2006] Schriek, Bernard: *Informatik mit Java. Eine Einführung mit BlueJ und der Bibliothek Stifte und Mäuse*. Werl : Nili-Verlag, 2006 ( Band II). – ISBN 3–00–019637–4. – Kapitel 1–4 (von 11):  
<http://www.mg-werl.de/sum/00P-Buch2.pdf> – geprüft: 17. Juni 2008
- [Schriek 2007] Schriek, Bernard: *Informatik mit Java. Eine Einführung mit BlueJ und der Bibliothek Stifte und Mäuse*. Werl : Nili-Verlag, 2007 ( Band III). – ISBN 978–3–00–022995–4. – Kapitel 1–4 (von 9):  
<http://www.mg-werl.de/sum/00P-Buch3.pdf> – geprüft: 17. Juni 2008