

# *Lernzielgraphen und Lernzielerfolgsanalyse*

Markus Schneider  
Institut für Informatik  
Technische Universität München  
[markus.schneider@in.tum.de](mailto:markus.schneider@in.tum.de)



## Aufbau

### **Inhaltsproblematik: Aktuelle Situation an Hochschulen aus didaktischer Sicht**

- *Welche Lernziele werden an Hochschulen aktuell gesetzt: Exemplarischer Überblick*

### **Didaktische Klassifizierungen der Informatik:**

- *Fundamentale Ideen, Great Principles*

### **Pragmatischer Zugang über eine Lernzielanalyse**

#### **Lernzielanalyse in der Praxis**

- *Bestimmung der Lernziele über Klausuraufgaben*
- *Exemplarische Lernzielfindung*
- *Lernpsychologische Schwierigkeitsbewertung*
- *Lernzielabhängigkeitsgraph*

#### **Lernziele und Klausurresultate**

- *Korrelierung von Klausurresultaten und Lernzielgraph*
- *Exemplarische Vorstellung der Methode zur Lernzielerfolgsanalyse*



## Wird mit dem Bachelor-/Masters wirklich alles anders ?

### Situation im Bachelor-/ Mastersystem:

- *Für die einzelnen Module werden mehr oder weniger detaillierte inhaltliche Vorgaben gemacht:*
  - [TU München](#)
  - [TU Berlin](#)
  - [TU Braunschweig](#)
  - [RWTH Aachen](#)

### Folgerungen:

- *Die Bachelor-Module der einzelnen Hochschulen unterscheiden sich erheblich*
- *Es steht zu befürchten, dass ein bundesweiter Konsens bezüglich einer einheitlichen Modulgestaltung nur schwer umzusetzen ist*
- *Einen inhaltlichen Konsens über das, was ein Student nach dem 1. Studienjahr wissen sollte, gibt es nicht.*



# Bachelor-System an der TU-München

Modulverantwortlicher:	Prof. Dr. Florian Matthes
	<p>In den Modulen IN0001 und IN0003 werden folgende Inhalte behandelt, die Aufteilung auf die Module wird vom Dozenten individuell festgelegt:</p> <ul style="list-style-type: none"><li>• Information und ihre Repräsentation</li><li>• Einfache Datenstrukturen: Atome und Sequenzen: Zeichen und Zeichenreihen, Zahlen</li><li>• Definition des Algorithmus über den Funktionsbegriff:<ul style="list-style-type: none"><li>◦ Vom Funktionsbegriff zum Begriff des Algorithmus</li><li>◦ Komposition von Funktionsaufrufen ohne Rekursion</li><li>◦ Rekursive Algorithmen</li></ul></li><li>• Algebraische Modellierung von Algorithmen und Datenstrukturen<ul style="list-style-type: none"><li>◦ Beispiele: Sortieren, Sequenzen von Atomen, bzw. Sequenzen</li><li>◦ Mustersuche</li><li>◦ Rekursive Datenstrukturen: Listen und Bäume</li><li>◦ Typisierung, parametrischer Polymorphismus</li></ul></li></ul>

[Zurück](#)



# Bachelor-System an der TU-Berlin

LP	Empfohlener Studienverlauf des Bachelor-Studiums				
1. Sem. 29 LP	TechGI 1 Digitale Systeme 6 LP	MPGI 1 Algorithmische und funktionale Lösung diskreter Probleme 9 LP	TheGI 1 Grundlagen und algebraische Strukturen 8 LP	Lineare Algebra für Ingenieure 6 LP	
2. Sem. 29 LP	TechGI 2 Rechnerorganisation 6 LP	MPGI 2 Datenstrukturen und Algorithmen im imperativen Stil 9 LP	TheGI 2 Automaten und Komplexität 6 LP	Analysis I für Inge- nieure 8 LP	
3. Sem. 32 LP	TechGI 3 System- programmierung 6 LP	MPGI 4 Softwaretechnik 12 LP	MPGI 3 Praxis der Pro- grammentwicklung 6 LP	TheGI 3 Logiken und Kalküle 6 LP	MA Analysis II für Ingenieure 8 LP
4. Sem. 30 LP	TechGI 4 Rechnernetze und Verteilte Systeme 6 LP		MPGI 5 Datenbanksysteme 6 LP	TheGI 4 Spezifikation und Semantik 6 LP	Stochastik für Informatiker 6 LP
5. Sem. 30 LP	Wahlpflichtmodule des Fachstudiums 21-24 LP Softwaretechnik oder Kommunikationstechnik		Anwendungsfach 12-15 LP		Grundlagen des Managements 6 LP
5. Sem. 30 LP			Bachelorarbeit 12 LP		Informatik und Gesellschaft 6 LP

[Zurück](#)



# Bachelor-System an der TU- Braunschweig

Bereich	1. Sem
Informatik	Programmieren I 4 L
	Technische Informatik I 4 L
	Theoretische Informatik I 4 L
	Algorithmen und Datenstrukturen I 8 L
	tage

Modulbezeichnung: <b>Programmieren I</b>		Modul-Nr. Bachelor: <b>INF1100</b>	
		Master: -	
Dauer: 1 Semester	Turnus: jährlich (WiSe)	Semester: 1.	Leistungspunkte: 4
Lehrveranstaltung: <ul style="list-style-type: none"> <li>• Programmieren I</li> <li>• Übung zu „Programmieren I“</li> <li>• Zusätzlich werden praktische Übungen in kleinen Gruppen angeboten.</li> </ul>		Lehrform:	SWS:
		Vorlesung	2
		Übung	1
Zuordnung Gebiet: Informatik Pflicht			
Qualifikationsziele: <ul style="list-style-type: none"> <li>• Nach Abschluss dieses Moduls besitzen die Studierenden grundlegende Kenntnisse der objektorientierten Programmierung sowie der Sprache Java.</li> <li>• Sie sind in der Lage, kleine Programme selbstständig zu entwickeln.</li> </ul>			
Lehrinhalte: <ul style="list-style-type: none"> <li>• Grundlagen der imperativen und objektorientierten Programmierung anhand der Sprache Java</li> <li>• rekursive Methoden</li> <li>• Zuverlässigkeit von Programmen</li> </ul>			

[Zurück](#)



er  
**Inform:**

# Bachelor-System an der RWTH-Aachen (Praktische Informatik)

Literatur	Folien und Skripte zur Vorlesung sowie z.B. folgende Bücher: <ul style="list-style-type: none"><li>• K. Echte &amp; M. Goedicke: <i>Lehrbuch der Programmierung mit Java</i>, dpunkt Verlag, 2000</li><li>• R. Bird: <i>Introduction to Functional Programming Using Haskell</i>, Prentice Hall, 1998</li><li>• W. F. Clocksin &amp; C. S. Mellish: <i>Programming in Prolog</i>, Springer, 2003</li></ul>
Lernziele	Erwerb der folgenden Kenntnisse und Fähigkeiten: <ul style="list-style-type: none"><li>• Beherrschung der wesentlichen Konzepte imperativer und objektorientierter Programmiersprachen sowie wichtiger Programmier Techniken in diesen Sprachen</li><li>• Kenntnis der Programmierkonzepte logischer und funktionaler Programmiersprachen</li><li>• Kenntnis grundlegender Datenstrukturen und ihrer Realisierung in verschiedenen Programmierparadigmen</li><li>• Fähigkeit zur selbständigen Entwicklung kleinerer Programme und ihrer Dokumentation unter Beachtung üblicher Programmierkonventionen</li><li>• Kenntnis grundlegender Beschreibungsformen für Programmiersprachen</li><li>• Grundkenntnisse der Programmverifikation</li></ul>

[Zurück](#)



# Welche Strukturierungen der Informatik sind bekannt

## „Fundamentale Ideen“ der Informatik (Schwill 1993)

- *Konzeptsammlung in hierarchischer Struktur*
- *Probleme*
  - Bei manchen Ideen erscheint die Fundamentalitätseigenschaft zweifelhaft,
  - Die Ideen an sich sind für die konkrete Anwendung in den Unterrichtsprozess zu abstrakt und allgemein formuliert

## „Great principles“ (Peter Denning 2003)

- *Aufteilung in „Great Principles of Computing“, „Computing Practices“, „Core technologies“*
- *Kein konzeptuelles Netzwerk, nur Mengen von Konzepten*
- *Great Principles noch nicht endgültig festgelegt*
- *Große Ähnlichkeit mit den fundamentalen Idee*



## Wo stehen wir?

### **Einführungsvorlesungen (im nationalen Rahmen)**

- *Es existieren verschiedene Schulen/Lehrzyklen mit sehr unterschiedlichen Lehrphilosophien*
- *Auch nach Einführung der Bachelor/Masterstudiengänge scheint sich das nicht geändert zu haben*

### **Konzeptuelle Strukturierungen der Informatik**

- *(Zu) Abstrakt formuliert: Zuordnung von Vorlesungsinhalten zu Konzepten im allgemeinen nur sehr schwer durchführbar*
- *Zum Teil fehlende empirische Grundlage*
- *Inhalte von Einführungsvorlesungen lassen sich nicht immer in die vorhandenen Strukturierung einordnen*



## Lernziele: Ein alternativer Weg

### Voraussetzung

- *Detaillierte inhaltliche Analyse der häufigsten Lehrphilosophien im Hinblick auf die zugrunde liegenden Lernziele!*

### Inhaltliche Analyse durch Lernzielanalyse:

- *Arbeitshypothese: Klausuren repräsentieren die wesentlichen Lernziele einer Vorlesung*
- *Der Lehrerfolg in den Lernzielen kann durch Analyse der Klausurresultate ermittelt werden*
- *Lernziele lassen sich hinsichtlich ihrer Schwierigkeitsgrades lernpsychologisch objektiv klassifizieren.*



# Lernzieltaxonomie nach Anderson/Krathwol

The Knowledge Dimension	The Cognitive Process Dimension					
	Remember	Understand	Apply	Analyze	Evaluate	Create
Factual Knowledge						
Conceptual Knowledge						
Procedural Knowledge						
Meta-cognitive Knowledge						



# Lernziele und Lernzielerfolgsanalyse: Wozu?

## **Schwierigkeitsanalyse der Klausuraufgaben**

- *Lassen sich objektive Kriterien für die Schwierigkeit einer Problemstellung angeben?*

## **Wie entwickelt sich die Leistung der Studierenden inhaltsspezifisch in**

- *in den ersten Jahren des Studiums ?*
- *abhängig von der Vorlesungsstrategie ?*

## **Lernziele unterschiedlicher Lehrstrategien**

- *Welche Lernziele werden in einführenden Vorlesungen zur Informatik gesetzt?*
- *Lässt sich aus der Menge dieser Lernziele ein gemeinsamer Katalog von Inhalten definieren? (Empirische Definition eines begrifflichen Netzes)*

## **Lernzielerfolgsanalyse**

- *Individuell: Gezielte Förderung der Studierenden*
- *Kollektiv: Anpassung der Lehre an die didaktische Situation*

## **Wie lassen sich diese Methoden auf die Schule übertragen ?**



# Beispiel: Lernpsychologische Lernzielkategorisierung

## Lernziel (Bachelor-Modul der RWTH-Aachen):

*„Beherrschung der wesentlichen Konzepte imperativer und objektorientierter Sprachen, sowie wichtiger Programmier Techniken in diesen Sprachen“*

## Zerlegung

- Wesentliche **Konzepte imperativer Sprachen beherrschen**
- Wesentliche **Konzepte objektorientierter Sprachen beherrschen**
- Wichtige **Programmier Techniken** imperativer Sprachen **beherrschen**
- Wichtige **Programmier Techniken** objektorientierter Sprachen **beherrschen**

Konzeptuelles Wissen

Prozedurales Wissen

= anwenden



# Beispiel: Lernpsychologische Lernzielkategorisierung

## Legende:

- Faktenwissen
- Konzeptuelles Wissen
- Prozedurales Wissen
- Ist Vorauss. für
- Ist Spezialisierung von

Imperatives Sprachvokabular beherrschen

Beschreibungstechniken objektorientierter Modellierung beherrschen

Konzepte imperativer Sprachen beherrschen:  
 Konzept der Zuweisung beherrschen  
 Referenzkonzept beherrschen  
 Das Konzept der Reihung beherrschen  
 Konzept der Schleife beherrschen  
 Konzept der bedingten Anweisung beherrschen  
 .....

Konzepte objektorientierter Modellierung beherrschen:  
 Klassenkonzept beherrschen  
 Aggregationskonzept beherrschen  
 Vererbungskonzept beherrschen  
 Konzept abstrakter Klassen/ Schnittstellen beherrschen  
 .....

Programmiertechniken imperativer Sprachen beherrschen:

Modellierungstechniken objektorientierter Sprachen beherrschen:

Programmiertechniken objektorientierter Sprachen beherrschen:



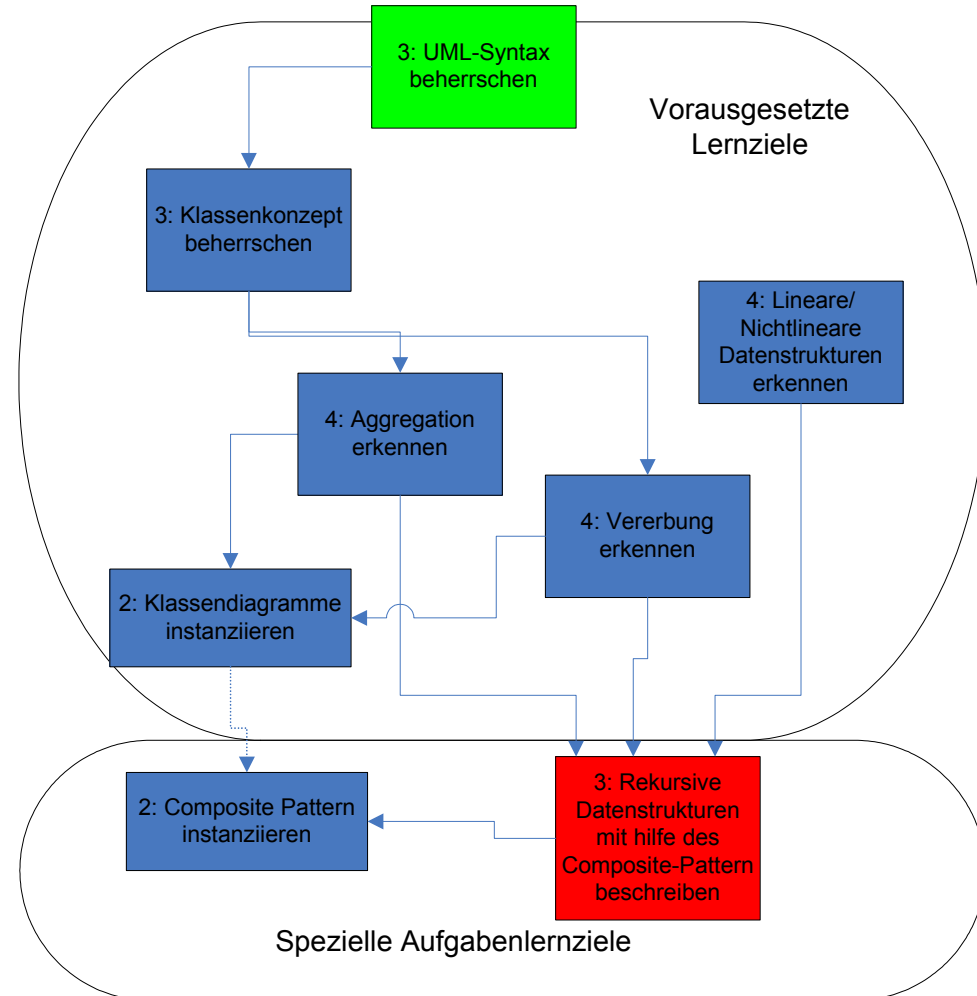
# Klausuraufgabenanalyse am Beispiel der Abschlussklausur zu Info I vom WS 2000

## Abschlussklausur: Einführung in die Informatik I vom WS 2000/2001

- Objektorientierte Modellierung (A/L)

### Legende:

- Faktenwissen
- Konzeptuelles Wissen
- Prozedurales Wissen
- Ist Vorauss. für (A. Staller)
- Ist Spezialisierung von

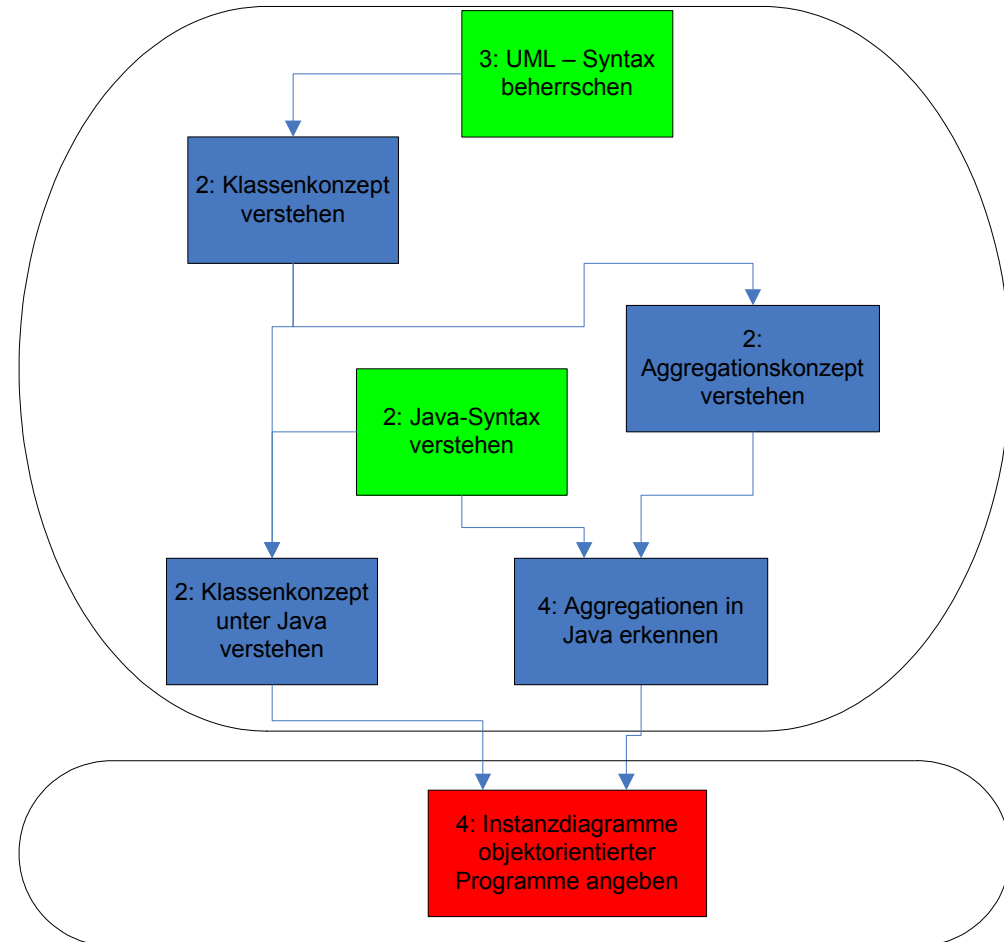




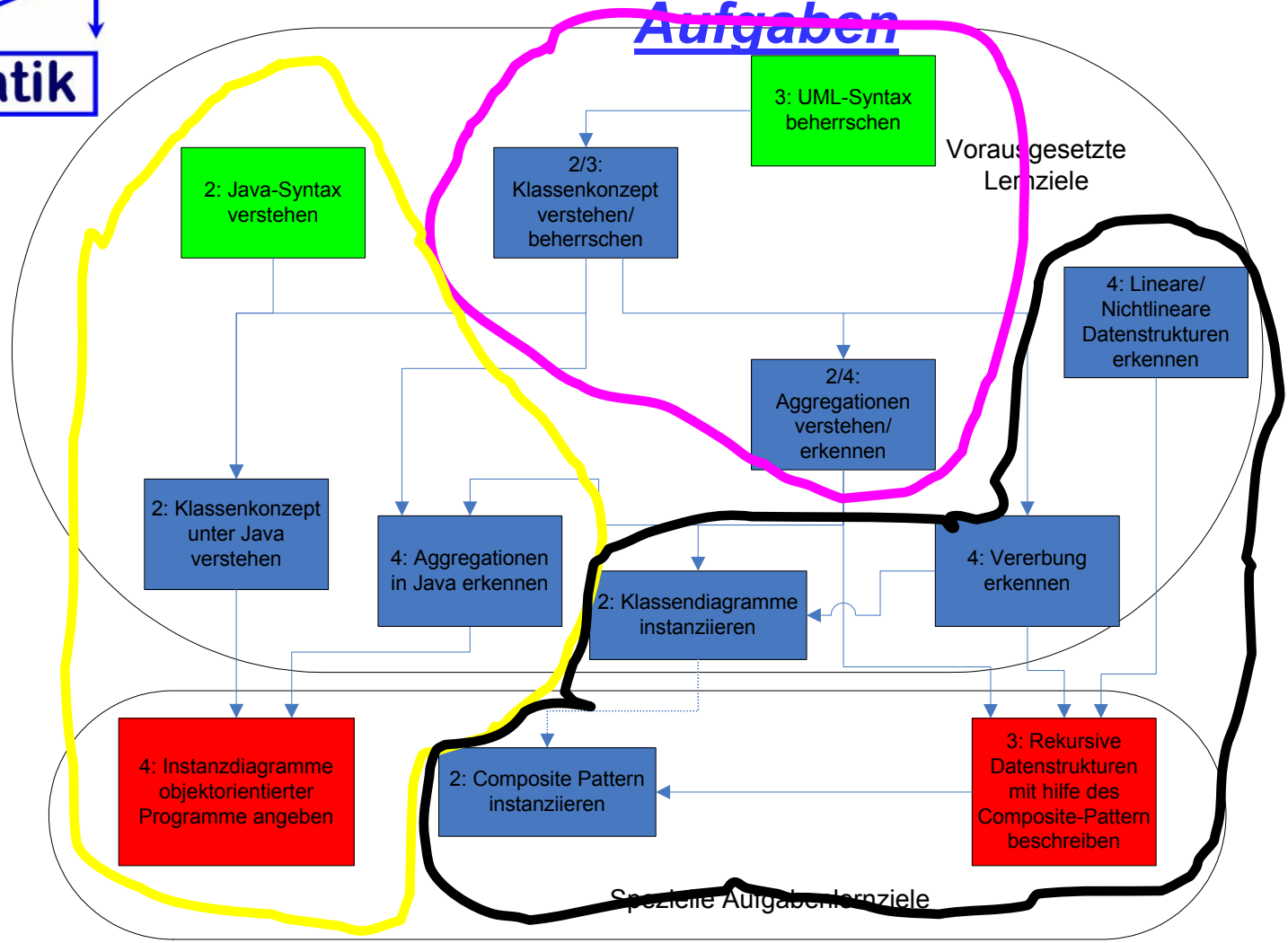
# Klausuraufgabenanalyse am Beispiel der Abschlussklausur zu Info I vom WS 2000

## Abschlussklausur: Einführung in die Informatik I vom WS 2000/2001

- Referenzgeflecht /  
Instanzdiagramm ([A/L](#))



# Abschlussklausur zu Info I vom WS 2000: Lernzielgraph mehrerer Aufgaben





# Abschlussklausur zu Info I vom WS 2000: Lernzielgraph mehrerer Aufgaben

## **Gemeinsame Lernziele:**

- UML-Syntax beherrschen
- Aggregationen verstehen/erkennen
- Klassenkonzept verstehen

## **Spezifische Lernziele der Modellierungsaufgabe:**

- Rekursive Strukturen erkennen
- Klassendiagramme/Composite-Pattern instanziiieren
- Rekursive Strukturen mit dem Composite Pattern beschreiben
- Vererbung erkennen

## **Spezifische Lernziele der Aufgaben zum Referenzgeflecht:**

- Java-Syntax beherrschen
- Klassenkonzept unter Java verstehen
- Instanzdiagramme von OO Programmen instanziiieren
- Aggregationen in Java erkennen



# Lernpsychologische Schwierigkeitsstufen

- **Schwierigkeit:** Wird jeweils für Faktenwissen, konzeptuelles und prozedurales Wissen getrennt angegeben
- **Flaschenhalsprinzip:** Schwierigkeit ergibt sich aus dem Maximum der Schwierigkeiten der vorausgesetzten Lernziele und der Schwierigkeit des Aufgabenzieles selbst

## Objektorientierte Modellierung

- *Faktenwissen: Stufe 3*
- *Konzeptuelles Wissen: Stufe 4*
- *Prozedurales Wissen: Stufe 3*

## Instanzdiagramm/Referenzgeflecht

- *Faktenwissen: Stufe 3*
- *Konzeptuelles Wissen: Stufe 4*
- *Prozedurales Wissen: Stufe 4*

Fazit: Rein lernpsychologisch besitzen die Aufgaben eine ähnliche Schwierigkeit. (Experteneinschätzung). → Ein Student, der alle Lernziele erreicht hat, müsste somit vergleichbare Leistungen in beiden Aufgaben erzielen!



## *Klausurresultate: Standardgruppen*

	<b>Modellierung</b>	<b>Referenzgeflecht</b>
<b>Alle (703)</b>	<b>50,85 %</b>	<b>33,91 %</b>
<b>Frauen (132)</b>	<b>47,92 %</b>	<b>23,96 %</b>
<b>Männer (567)</b>	<b>51,57 %</b>	<b>36,20 %</b>
<b>Mathematik Diplom (21)</b>	<b>59,52 %</b>	<b>54,27 %</b>
<b>Informatik Diplom (570)</b>	<b>52,68 %</b>	<b>35,46 %</b>
<b>Informatik Bachelor (66)</b>	<b>35,61 %</b>	<b>21,59 %</b>



## Lernzielerfolgsanalyse nach speziellen Gruppen

**Hypothese: Studenten, die beide Aufgaben mangelhaft (< 40 %) gelöst haben, haben die beiden Aufgaben gemeinsamen Lernziele nicht erreicht**

**→ ca. 31 % der Studenten verstehen weder das Klassen- noch das Aggregationskonzept und beherrschen die UML Syntax nicht (Gruppe A)**

**Hypothese: Studenten, die genau eine der beiden Aufgaben mangelhaft gelöst haben, haben die Lernziele, die ausschließlich dieser Aufgabe zugeordnet sind, nicht erreicht**

**(Modellierung > 40 % und Referenzgeflecht < 40 %)**

**→ ca. 33 % der Studenten verstehen ein Java-Programm nicht, oder können das Klassenkonzept nicht in Java umsetzen, obwohl sie das Klassen- und Aggregationskonzept verstanden haben (Gruppe B).**



## Lernzielerfolgsanalyse nach speziellen Gruppen

(Modellierung < 40 % und Referenzgeflecht > 40 %) → ca. 4 % der Studenten können rekursive Strukturen nicht mit Hilfe des Comp.-Pattern darstellen, obwohl sie das Klassen- und Aggregationskonzept verstanden haben, UML-Syntax beherrschen und das Klassenkonzept in Java umsetzen können (Gruppe C).

Hypothese: Studenten, die beide Aufgaben mit mehr als 40 % gelöst haben, haben die vorausgesetzten Lernziele beider Aufgaben erreicht:

→ ca. 32 % der Studenten haben die vorausgesetzten Lernziele erreicht (Gruppe D).



Wie haben die verschiedenen Gruppen in den verschiedenen Klausuraufgaben im Mittel abgeschnitten ?

	A	B	C	D
Signaturen und Terme	63,78%	81,62%	87,50%	90,17%
Boolesche Funktionen	35,74%	50,43%	72,99%	76,74%
Markov-Algorithmus	31,51%	58,17%	68,53%	79,56%
Rekursive Programmierung	28,37%	51,01%	78,45%	87,56%
Sortieren von Sequenzen von Zahlen	21,72%	37,87%	60,34%	72,50%
Reihungen	5,76%	14,37%	33,19%	55,50%
Binärbäume	8,42%	22,39%	50,69%	62,84%
Modellierung	17,15%	61,22%	21,98%	76,00%
Referenzgeflecht/Instanzdiagramm	6,10%	9,83%	70,69%	80,78%



# Mögliche weitere Vorgehensweisen und deren Verwendung

## **Optimierung der Lernzielerfolgsanalyse**

- *Formalisierung des Analyseverfahrens*
- *Kombination mehrerer Aufgaben: Aussagen über einzelne Lernziele*

## **Analyse von Vorlesungsstrategien**

- *Vergleich, inwieweit eine bestimmte Lehrstrategie dem Erreichen eines bestimmten Lernzieles förderlich ist*
  - Umfangreiche Lernzielerfolgsanalyse
- *→ Optimierung der Lehrstrategie*

## **Empirische Definition der zentralen Konzepte der Informatik**

- *Erstellung des Lernzielgraphen repräsentativer Lehrstrategien*
- *Definition der zentralen Konzepte über die Menge gemeinsamer Lernziele*
- *Empirische Definition eines Begriffsnetzes der Informatik*



# Einsatzmöglichkeiten und mögliche weitere Vorgehensweisen

## Gezielte Förderung von Studenten

- *System zur zielgerichteten Analyse der Wissenslücken eines Studierenden*
- *Fokussierung der Vorlesungsstrategie auf die Wissensdefizite der Studenten*
- *Einrichtung spezieller Förderungsgruppen*

## Lernzielanalyse im Schulfach Informatik

- *Inhaltliche Analyse*
  - Jahrgangsstufe 6/7: Objektorientierte Analyse und Einführung in das algorithmische Denken
  - Jahrgangsstufe 9: Funktionales Modellieren, Datenmodellierung
  - Jahrgangsstufe 10: Objektorientierte Ablaufmodellierung
- *Langzeitstudie:*
  - Lernzielerfolg im Laufe des „gymnasialen Schülerlebens“
  - Ausnutzen aktueller breit angelegter Bestrebungen zur Etablierung einer zentralen „Testkultur“ an bayerischen Gymnasien