Einführung die Programmierung – Erfahrungen aus 12 Jahren

Hochschule Wismar

Fakultät für Wirtschaftswissenschaften
Uwe Lämmel

www.wi.hs-wismar.de/uwe.laemmel

Gliederung

- Einführung in die Programmierung
- Java-Konzepte
- Lehr-Konzepte
- Klausuren
- Fazit

Module „Einführung in die Programmierung"

- 4 SWS: 2 V / 2 L
- Prüfung
  - Klausur 120 min
  - Papier und Stift
- Der Traum
  - Vorlesung: Studenten aufmerksam und fragen
  - Labor: intensive Diskussion, Programmierübung
  - zu Hause: selbstständiges Programmieren

HOCHSCHULE WISMAR UNIVERSITY OF TECHNOLOGY, BUSINESS AND DESIGN

Gliederung

- Einführung in die Programmierung
- **Java-Konzepte**
- Lehr-Konzepte
- Klausur
- Fazit

Java-Konzepte

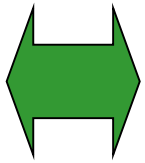
- Objekte und Klassen
- Sammlungen
- Keller, Schlange, Tabelle, Baum
- Suchen und Sortieren
- Vererbung

Module Inhalt

- Klassen und Objekte
  - Attribute, Konstruktoren, get/set-Methoden
- Anweisungen und Datentypen
  - einfache Anweisungen, if-Anweisung
  - einfache Datentypen: int, char, double, boolean
  - Objekt-Typen
- Sammlungen
  - ArrayList, Array, Schleifen (for-each, while, for, Iterator)
- Container
  - Keller, Schlange, Baum
  - Implementierungen mit Array, ArrayList oder rekursiv
- Algorithmen
  - Suchen, Sortieren
- Vererbung

Paradigma

Welt-Sicht
Subjekt + **Aktion**



Computer
Daten + **Algorithmen**

BlueJ

- objekt-orientierte Perspektive
- reines Entwickeln von Klassen
- Objekte: interaktiv erzeugen
- Zustand eines Objektes einsehbar
- keine main-Methode notwendig
- keine E/A-Operationen erforderlich
- keine bunten Knöpfe,
  die hunderte von Programmzeilen generieren

www.bluej.org

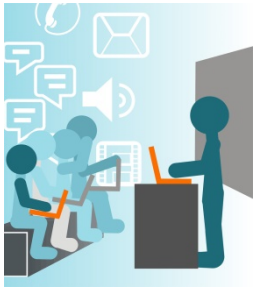
Gliederung

- Einführung in die Programmierung
- Java-Konzepte
- **Lehr-Konzepte**
- Klausur
- Fazit

Ziele

- Studenten
  - ... sollen programmieren lernen
  - ... sollen die Prüfung bestehen
- wir “predigen”
  - kontinuierliches Üben
  - Problem lösen
  - Kommunikation

Lehr-Konzept 1



- **traditionelle Vorlesung**
  - Konzepte vorstellen
  - Beispiele programmieren
- **wöchentliche Übung**
  - Übungsaufgaben werden individuell vorgestellt
- **Studenten sammeln Punkte**
  - Punkte werden in die Klausur einbezogen

Lehr-Konzept 1 – Erfahrung

\+ persönliches Feedback

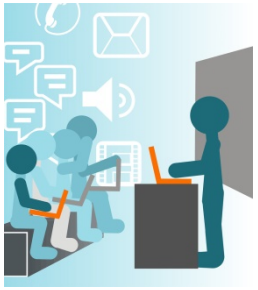
– Thema der Aufgabe 10-14 Tage alt:
in der Vorlesung bereits andere Themen

– Labor-Zeit nur für Präsentation der Ergebnisse

o Studenten, die Übungsaufgaben bearbeiten,
benötigen die Punkte nicht für die Klausur

Lehr-Konzept 2

- **klassische Vorlesung**
- **klassische Labor-Übung**
  - programmieren unter Anleitung
- **KEINE wöchentlichen Aufgaben**



Erfahrungen

\+ mehr Zeit im Labor für Diskussion

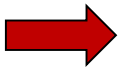
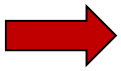
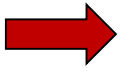
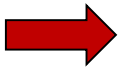
– weniger Studenten besuchen LV

– schlechtere Ergebnisse in der Klausur?

Lehr-Konzept 3

- **Wöchentliche Aufgabe**
  - Programmieraufgabe
  - Antworten auf Zusatzfragen erforderlich
  - Hochladen in Stud.IP
    Abgabe etwa einen Tag vor Vorlesung
- **Vorlesung: Besprechen der Lösungen**
- **Studenten sammeln Punkte**
  - werden in Klausur einbezogen

Fragen

```
/**
 * Write a description of class Student here.
 *
 * @author Author
 * @version DD.MM.YY
 *
 * @Schwierigkeitsgrad der Aufgabe:
 *    0-einfach, 1-lösbar, 2-schwierig/viel Zeit, 3-unlösbar
 *
 * @Besonders schwierig war für mich ...
 * ...
 *
 * @Nicht verstanden habe ich ...
 * ...
 *
 * @Ich habe folgende Frage(n):
 * ...
 *
 */ public class Student {
    // instance variables
    private int x;
...
```

Lehr-Konzept 3 – Jitt?

- vor der Vorlesung
  - Analyse der Fragen und Kommentare
  - Bewerten der Programmieraufgabe

Just in time teaching ?

- **in der Vorlesung**
  - Auswertung der Aufgabe ➔ 50% – 80%
  - kurze Einführung in neue Aufgabe und ggf. Konzepte

- Diskussion der Aufgabe
  - Originaltext (anonym) wird verwendet
    - Besonders schwierig ...
    - Nicht verstanden ...
    - Fragen
  - Beispiellösung

Flipped classroom ?

Lehr-Konzept 3 – Erfahrung

\+ Studenten drücken ihre Probleme aus

\+ schnelle Rückmeldung

\+ Selbststudium wird gefördert

\+ Laborzeit zum Üben

o Studenten, die Aufgaben bearbeiten, bestehen auch ohne Zusatzpunkte!

– zeitaufwändig (für mich)

Gliederung

- Einführung in die Programmierung
- Java-Konzepte
- Lehr-Konzepte
- **Klausur**
- Fazit

Klausur

- konstante Struktur
- 7 Aufgaben
  - Bauplan-Kasse
  - Theorie-Frage
  - Keller, Schlange
  - rekursive Programmierung (Baum)
  - Sortieren
  - main-Methode
  - Vererbung
- 100 Punkte
  - ≥ 50% ➔ bestanden

Klausur-Ergebnisse

	Jahr	Anzahl	Mittelw	Note
klassische Vorlesung wöchentliche Aufgaben	2006	52	55.0	3,6
	2007	32	50.3	3,6
klassische Vorlesung keine Aufgaben	2008	51	51.2	3,4
	2009	33	56.9	3,3
	2010	39	60.5	3,3
	2011	31	52.0	3,6
	2012	28	42.8	4,2
	2013	15	54.1	3,6
	2014	39	58.5	3,6
jitt Vorlesung wöchentliche Aufgaben	2015	31	75.1	2,5
	2016	24	69.8	2,9
	2017	23	63.3	2,9

Einfluss Übungsaufgaben

	Mittel						Durchfall				Einfluss
	Anz	ÜA	Kl	Note	+ÜA	Note	5,0	%	5,0	%	ÜA
2006	52	4,6	50,6	3,9	55,0	3,6	24	46,2%	21	40,4%	5,8%
2007	32	6,3	44,4	3,9	50,3	3,6	17	53,1%	14	43,8%	9,4%
2008	51		51,0	3,5	51,2	3,4	19	37,3%	18	35,3%	2,0%
2009	33		56,9	3,3	56,9	3,3	13	39,4%			
2010	39		60,5	3,3	60,5	3,3	12	30,8%			
2011	31		52,0	3,6	52,0	3,6	13	41,9%			
2012	28		42,8	4,2	42,8	4,2	19	67,9%			
2013	15		54,1	3,6	54,1	3,6	6	40,0%			
2014	39		58,5	3,6	58,5	3,6	15	38,5%			
2015	31	5,5	69,8	2,8	75,1	2,5	3	9,7%	2	6,5%	3,2%
2016	24	6,7	63,8	3,2	69,8	2,9	8	33,3%	8	33,3%	0,0%
2017	23	6,3	57,5	3,3	63,3	2,9	8	34,8%	8	34,8%	0,0%

Fazit

year	students	points	mark
2006	52	55.0	3,6
2007	32	50.3	3,6
2008	51	51.2	3,4
2009	33	56.9	3,3
2010	39	60.5	3,3
2011	31	52.0	3,6
2012	28	42.8	4,2
2013	15	54.1	3,6
2014	39	58.5	3,6
2015	31	75.1	2,5
2016	24	69.8	2,9
2017	23	63.3	2,9

jitt Vorlesung

- Erster Gedanke
  - Jitt hilft!
- Dann:
  - andere Einflussfaktoren?

- **Gruppendynamik?**
- einige wenige bestimmen Lern-Athmosphäre
- Kommunikation
- vielleicht hilft jitt, eine gute Atmosphäre zu schaffen

Danke

Uwe Lämmel
Hochschule Wismar
Germany
www.hs-wismar.de/uwe.laemmel

Just-in-time teaching (Jitt)

- Online questions and exercise BEFORE classroom
- Discussion of student results in classroom

Jitt <> traditional

- First, having completed the web assignment very recently, students enter the classroom ready to participate...
- Second, students have a feeling of ownership because classroom activities are grounded in their own understanding of the relevant issues

Novak &Patterson: An introduction to Just-in-Time Teaching,
in: Simkins & Maier: Just-in-time teaching, Stylus Publishing 2010, p.7

Just-in-time teaching

- yield a rich set of students responses for classroom discussion.
- encourage students to practice prior knowledge and experience
- require an answer that cannot easily be looked up
- require that students formulate a response ... in their own words
- contain enough ambiguity to require the student to supply some additional information not explicitly given in the question

Novak &Patterson: An introduction to Just-in-Time Teaching,
in: Simkins & Maier: Just-in-time teaching, Stylus Publishing 2010, p.7