

---

# Rückblick

---

# Klischees bestätigt

- überfüllter Hörsaal
- chaotische Organisation
- frontale Massenlehre statt persönlicher Betreuung
- ewig auf Übungs- und Testergebnisse warten
- wir sind die Versuchskaninchen

Aber (wie in jedem Semester):

- „Im nächsten Semester machen wir es besser!“

---

## Was neu war

- Trennung zwischen direkt vermittelbarem Wissen (PK) und nur durch Selbermachen erlernbarem Wissen (PP)
- viel kompliziertes Wissen gleich am Anfang
- mehrfache Zugänge aus verschiedenen Perspektiven

Hat es sich ausgezahlt?

- noch keine echten Ergebnisse
- aber deutliche Hinweise auf Qualitätssteigerung

---

## Was auffällt

- Großteil der Studierenden kann mit komplizierten Themen recht gut umgehen
- einige kleine Fehler sind schwer zu beseitigen
- wer sich wirklich bemüht, kann es schaffen
- auch erfahrene Leute wissen nicht alles

---

# Ausblick

---

# PK geschafft! Was nun?

- Algorithmen und Datenstrukturen
- Objektorientierte Modellierung
- Software Engineering
- Programmierparadigmen  
(Objektor. Prog.techniken, Funktionale Programmierung)
- . . .

---

# Kann ich jetzt Programmieren?

- in PK nur Grundlagen vermittelt
- Fähigkeiten entwickeln sich langsam mit der Praxis
- Programmieren lernt man in anderen LVAs nicht, aber viele LVAs entwickeln Programmierfähigkeiten weiter
- Erfahrungen in einer Sprache lassen sich leicht auf andere Sprachen übertragen
- man hat nie ausgelernt

---

# PK nicht geschafft :- ( – was nun?

- Ursache ermitteln
  - Anfang verschlafen oder Aufwand unterschätzt
  - zeitliche und organisatorische Probleme
  - zu wenig Motivation, zu wenig bemüht
  - “Will ich wirklich Informatik machen?”
- Ursache vermeiden oder es bleiben lassen
- zweite Chance nutzen, jetzt aber wirklich
- eine dritte Chance wird es wahrscheinlich nicht geben (formal schon, aber wegen STEOP nicht wirklich)



---

## Neuerungen in PK

- neuer Hörsaal
- Übung in Gruppen
- viele kleine Änderungen im Stoff
- kleine Änderungen bei Prüfungen
- aber im Großen und Ganzen bleibt alles beim Alten

---

# Vorschläge für die Ferien

---

# Sudokus lösen

- Motivation: Gehirntraining immer gut
- Vorgehensweise:
  - Regeln lernen
  - beispielhaft einige Sudokus lösen
  - dabei einen Lösungsansatz entwickeln (Abstraktion)
  - Lösungsansatz in Programm umsetzen
  - durch Automatisierung freigesetzte Kapazitäten sinnvoller nutzen, beispielsweise durch
    - \* Einbeziehung komplexerer Arten von Sudokus
    - \* Entwickeln eines Generators für Sudokus

---

## Wenn es langweilig wird . . .

- kleine Happen für Zwischendurch:  
Tik-Tak-Toe, Vier-gewinnt, Mastermind, . . .
- Mensch gegen Mensch rasch ausprogrammiert
- Mensch gegen Computer auch einfach machbar
- für die ganz große Langeweile:  
Computer gegen Computer spielen lassen

---

## Zeigen was man kann

- Problem: gute Programme können Freunde, Eltern, ... kaum überzeugen (die verstehen das nicht)
- Lösungsansatz: *Fraktale* berechnen und zeichnen (z.B. Ausschnitte aus Mandelbrot-Menge)
- Vorteil: rasch gemacht, wirkt immer gut
- Nachteil: man verfällt leicht dem Reiz der Bilder

---

# Schifahren & Co

- Problem: Simulation erzeugt nicht das richtige Feeling
- Ursache: „Um die reale Welt simulieren zu können, muss man sie kennen.“
- Lösungsansatz: Computer ausgeschaltet lassen
- Ziel: Schöne Ferien!

---

**Danke für Ihre Mitarbeit**