

PK

Ausblick

PK bald geschafft! Was nun?

Programmierpraxis abschließen

Algorithmen und Datenstrukturen

Objektorientierte Modellierung

Programmierparadigmen

(Objektorientierte Programmier Techniken, Funktionale Programmierung)

Software Engineering

Kann ich jetzt Programmieren?

in PK nur breit gestreute Grundlagen mit wenig Tiefe vermittelt

Programmierfähigkeiten entwickeln sich langsam mit der Praxis

Programmieren lernt man nur in PP und PK,
aber viele Lehrveranstaltungen 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, organisatorische, inhaltliche Probleme,
„Will ich wirklich Informatik machen?“

„Pech gehabt“ gilt nicht (sehr schlechte Ausrede)

Ursache vermeiden oder Informatik-Studium bleiben lassen

zweite Chance nutzen, jetzt aber wirklich,
denn eine dritte Chance wird es für viele nicht geben
(formal schon, aber durch sinkende Motivation stark erschwert)

Aufgabe: Was war PK und wie geht es weiter?

Such Sie in Gruppen zu zwei bis drei Personen Antworten auf diese Fragen:

1. Was waren die Lernziele von PK?
2. Inwieweit haben wir (persönlich) diese Lernziele erreicht?
3. Wo und wie werden wir (persönlich) das Gelernte einsetzen können?
4. Wie geht es für uns (persönlich) im Sommersemester weiter?

Zeit: 2 Minuten

Aufgabe: Was war gut/schlecht an PK?

Such Sie in Gruppen zu zwei bis drei Personen Antworten auf diese Fragen:

1. Was hat uns an PK gefallen?
2. Was sollte in Zukunft geändert werden?

Zeit: 2 Minuten

Hinweis: Bewertung der Lehrveranstaltung in TISS

PK im Sommersemester

Vorlesungen von **Anton Ertl** gehalten,
Übungen weiter von **Markus Raab** betreut

sicher einige Änderungen im Stoff und in der Organisation

es ist noch offen, wer Programmierpraxis betreuen wird

PK

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 sinnvoll nutzen, z.B.

komplexere Arten von Sudokus einbeziehen,

Generator für Sudokus entwickeln

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

Zeigen was man kann

Problem: gute Programme können Freunde und 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
(statt die Schönheit guten Codes zu bewundern)

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

kurzfristiges Ziel: Spaß haben

längerfristiges Ziel: mit Schwung in's nächste Semester

PK

**Viel Erfolg beim 3. Test
(und bei Bedarf beim Nachtragstest)**

PK

Danke für Ihre Mitarbeit

Schöne Ferien