

Programmierung

Prof. Dr. Michael Hanus
mh@informatik.uni-kiel.de, Tel. 880-7271, R. 706 / CAP-4

Fabian Reck
fre@informatik.uni-kiel.de, Tel. 880-7262, R. 701 / CAP-4

Institut für Informatik
Christian-Albrechts-Universität zu Kiel

WS 2011/2012

Einführung

Hörerkreis

Termine

Übungen

Prüfungsmodalitäten

Weitere Informationen zu den Studiengängen

Inhalt

Hörerkreis

- ▶ Bachelor Informatik (1-Fach)
- ▶ Bachelor Informatik (2-Fach)
- ▶ Bachelor Wirtschaftsinformatik
- ▶ Bachelor Mathematik mit Nebenfach Informatik
- ▶ ...

Nicht: Bachelor Physik mit Nebenfach Informatik im 1. Semester

- ▶ Informatik für Naturwissenschaftler
(Beginn: Montag, 24.10.11, 16:00 Uhr, CAP3-HS II)

Termine

Vorlesung: Montag, 10:15 – 12:00 Uhr [Alte Mensa gr.HS]
Freitag 8:15 – 10:00 Uhr, [Alte Mensa gr.HS]

Kleingruppenübungen:

Montag, 14:15 – 15:45 (WSP3 - R.1)

Montag, 14:15 – 15:45 (WSP3 - R.2)

Montag, 14:15 – 15:45 (WSP3 - R.3)

Montag, 16:15 – 17:45 (WSP3 - R.1)

Montag, 16:15 – 17:45 (WSP3 - R.2)

Dienstag, 12:15 – 13:45 (WSP3 - R.1)

Dienstag, 14:15 – 15:45 (WSP3 - R.1)

Dienstag, 14:15 – 15:45 (WSP3 - R.2)

Dienstag, 16:15 – 17:45 (WSP3 - R.1)

Mittwoch, 16:15 – 17:45 (WSP3 - R.1)

Anmeldung zu den Übungen

- ▶ Anmeldung zu den Übungen: **ab 22.10.2011, 10:00 Uhr** in der StudiDB

<http://www.informatik.uni-kiel.de/ifi/studium-lehre/studidb/>
 (auch www.informatik.uni-kiel.de
 ↪ Studium&Lehre ↪ StudiDB)

- ▶ Studierende mit Hauptfach Informatik: direkt anmelden
- ▶ Studierende anderer Fächer:
 zunächst (sorgfältige!) StudiDB-Account-Anmeldung (**heute!**),
 danach Anmeldung zu den Übungen
- ▶ Bildung von **Zweiergruppen** für die Bearbeitung und Abgabe
 (später, aber gemeinsam in einer Gruppe anmelden!)
- ▶ Anmeldung möglichst bis Montag, 24.10.2011, 10:00 Uhr

Übungsbetrieb (↔ Fabian Reck, fre@informatik.uni-kiel.de)

- ▶ Ausgabe der Übungsblätter: freitags (Vorlesung)
- ▶ Präsenzaufgaben und abzugebende Aufgaben
- ▶ Abgabe der Lösungen: (eine Woche später, 8 Uhr):
per WWW im “iLearn” System
(Pflicht bei Programmieraufgaben!),
evtl. auch in Papierform in der Vorlesung
- ▶ Beginn der Übungen: **Montag, 24.10.2011**

Rechnerübungen

- ▶ Grundausbildungspool
- ▶ Einführung in der 1. Übung
- ▶ auch zu Hause. . .

Modulprüfung:

- ▶ Abschlussklausur: (am Vorlesungsende)
- ▶ Teilnahmevoraussetzung: aktive Übungsteilnahme
 - ▶ Anwesenheitspflicht in den Kleingruppenübungen!
 - ▶ Vorbereitung der Präsenzaufgaben

Bonuspunkte:

- ▶ erreichte Übungspunkte können Abschlussklausur zu 20% verbessern
- ▶ Voraussetzung: $\geq 50\%$ der Übungen erfolgreich bearbeitet
- ▶ Beispiel: 100% aller Übungen erfolgreich bearbeitet
 \rightsquigarrow 20% der Klausurpunkte als Bonus hinzugefügt
- ▶ Beispiel: 50% aller Übungen erfolgreich bearbeitet
 \rightsquigarrow 10% der Klausurpunkte als Bonus hinzugefügt

Üben lohnt sich . . . und ist notwendig!

Einführendes Programmierpraktikum

(↔ Dr. Henning Schnoor)

- ▶ Pflicht für Studierende mit Studienziel „Bachelor“ (1-Fach)
- ▶ Beginn: ab 9.1.2012

2-Fach-Bachelor-Studierende

- ▶ 1. Fachsemester:
Programmierung + Mathematik für Informatiker A
- ▶ falls **2. Fach Mathematik**: statt „Math. für Informatiker A“:
„Digitale Systeme“
- ▶ Betreuung aller 2-Fach-Bachelor-Studierenden durch Prof.
Wilke

Zwei-Fächer-Teilstudiengang Informatik

Informationen im OLAT (in der Suchmaske nach „Zwei“ suchen!)

- ▶ Eintragung für den 2F-Inf-Treff (siehe unten)
- ▶ Eintragung für den 2F-Inf-Rundbrief (wichtige Informationen zum Teilstudiengang Informatik)
- ▶ erweiterter Foliensatz aus der Begrüßungsveranstaltung
- ▶ Sprechzeiten Studienberater

2F-Inf-Treff

- ▶ Di., 9.30–10.30, CAP 4, Raum 1011, ab 1.11.
- ▶ Hilfe in Mathematik A, Einf. in die Programmierung, ...
- ▶ Hilfe bei organisatorischen Schwierigkeiten

Vorlesungsetikette:

- ▶ Handys abschalten
- ▶ Laptop-Nutzung:
 - ▶ nur für die Vorlesung (Skript lesen, schreiben, programmieren, . . .)
 - ▶ keine Spiele, Facebook etc!
- ▶ Fragen erwünscht!

Inhalt

Generell:

- ▶ „Programme müssen geschrieben werden, damit Menschen sie lesen, und nur nebenbei, damit Maschinen sie ausführen.“
- ▶ „... Techniken, mit denen die geistige Komplexität großer Softwaresysteme unter Kontrolle gehalten werden kann.“
- ▶ „Wir halten Komplexität unter Kontrolle, indem wir **Abstraktionen** bilden.“

Literatur:

H. Klaeren, M. Sperber: Die Macht der Abstraktion,
Teubner, 2007 (44,95 Euro)

M. Felleisen, R.B. Findler, M. Flatt, S. Krishnamurthi:
How to design programs, MIT, 2001 (75,00 US\$)

H. Abelson, G.J. Sussman: Struktur und Interpretation von
Computerprogrammen, Springer, 2001 (32,95 Euro)

Skript zur Vorlesung:

wird über die Vorlesungswebseite zur Verfügung gestellt:

www.informatik.uni-kiel.de/~mh/lehre/Inf-Prog-WS11/

Kurzübersicht:

1. Grundbegriffe
2. Abstraktion mit Prozeduren
3. Abstraktion mit Daten
4. Modularität, Objekte, Zustände