

Programmierung

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

Jan Rasmus Tikovsky
jrt@informatik.uni-kiel.de, Tel. 880-7263, R. 702 / CAP-4

Institut für Informatik
Christian-Albrechts-Universität zu Kiel

WS 2014/2015

Einführung

Hörerkreis

Termine

Übungen

Prüfungsmodalitäten

Inhalt

Hörerkreis

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

Bitte im Umlaufzettel eintragen!

Termine

Vorlesung: Montag, 10:15 – 12:00 Uhr [Norbert-Gansel-Hörsaal]
Freitag 8:15 – 10:00 Uhr, [Norbert-Gansel-Hörsaal]

Kleingruppenübungen:

Montag, 12:15 – 13:45 (CAP4 - R.1304a)

Montag, 14:15 – 15:45 (LMS2 - R.Ü3)

Montag, 14:15 – 15:45 (CAP4 - R.1304b)

Montag, 16:15 – 17:45 (LMS2 - R.Ü3)

Dienstag, 12:15 – 13:45 (CAP4 - R.1304a)

Dienstag, 12:15 – 13:45 (LMS2 - R.Ü2)

Dienstag, 14:15 – 15:45 (LMS2 - R.Ü1)

Dienstag, 14:15 – 15:45 (LMS2 - R.Ü3)

Dienstag, 16:15 – 17:45 (CAP4 - R.1304a)

Anmeldung zu den Übungen

- ▶ Anmeldung zu den Übungen: **ab 25.10.2014, 10:00 Uhr** in der StudiDB
`http://www.inf.uni-kiel.de/de/studium/pruefungen/studidb/`
(auch `www.inf.uni-kiel.de` ↪ Quicklink „StudiDB“)
- ▶ **Vorher** stu-Account aktivieren und in der StudiDB registrieren (insbesondere Studierende anderer Fächer)
- ▶ Bildung von **Zweiergruppen** für die Bearbeitung und Abgabe (später, aber gemeinsam in einer Übungsgruppe anmelden!)
- ▶ Anmeldung möglichst bis Montag, 27.10.2014, 10:00 Uhr

Übungsbetrieb (↔ Jan Tikovsky, jrt@informatik.uni-kiel.de)

<https://www.informatik.uni-kiel.de/prog/mitarbeiter/jan-tikovsky/lehre/programmierung-14/>

- ▶ Ausgabe der Übungsblätter: freitags
- ▶ 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
- ▶ Account für “iLearn” wird später vergeben (**nicht anmelden!**)
- ▶ Beginn der Übungen: **Montag, 27.10.2014**

Rechnerübungen

- ▶ auf dem eigenen Rechner
- ▶ virtuell auf einem Ifl-Rechner:
Terminal-Server mit Software Client ThinLinc und verbinden
zu `isildur.informatik.uni-kiel.de`
`www.inf.uni-kiel.de/de/service/technik-service/dienste`
- ▶ im Ifl-Grundausbildungspool

↪ in der Übung nachfragen

Modulprüfung:

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

Bonuspunkte:

- ▶ erreichte Übungspunkte können Abschlussklausur zu 20% verbessern
- ▶ Voraussetzung: $\geq 50\%$ der Übungen erfolgreich bearbeitet und aktive Teilnahme an den Übungsgruppen
- ▶ Beispiel: 100% aller Übungen erfolgreich bearbeitet
↪ 20% der Klausurpunkte als Bonus hinzugefügt
- ▶ Beispiel: 50% aller Übungen erfolgreich bearbeitet
↪ 10% der Klausurpunkte als Bonus hinzugefügt

Üben lohnt sich ... und ist notwendig!

Einführendes Programmierpraktikum (Prof. Wilke)

Was?

- ▶ Pflicht für Studierende mit Studienziel „Bachelor“ (1-Fach)
- ▶ Anwendung der in „Programmierung“ erlernten Techniken
- ▶ Entwicklung etwas größerer Programme
- ▶ Arbeit in 2er-Teams

Wann?

- ▶ Blockveranstaltung in der vorlesungsfreien Zeit
- ▶ Start: 23. Februar 2015

Anmeldung?

- ▶ Ab November
- ▶ Weitere Informationen folgen. . .

Vorlesungsetikette:

- ▶ Keine Störungen
- ▶ Handys abschalten
- ▶ Laptop/Tablet-Nutzung:
 - ▶ nur für die Vorlesung (Skript lesen, schreiben, programmieren, . . .)
 - ▶ keine Spiele, Facebook etc!
- ▶ Beschweren Sie sich!
- ▶ Fragen erwünscht!
- ▶ Abschreiben in der Wissenschaft ist inakzeptabel!

Übungsetikette:

- ▶ siehe Vorlesungsetikette... +
- ▶ Vorbereitet in die Übung gehen:
 - ▶ Präsenzaufgaben
 - ▶ Verständnisfragen (Vorlesung / nächste Übung)
- ▶ Nutzen Sie das Angebot!
 ~> Sprechstunden
- ▶ Schauen Sie sich die Korrekturen an!
 (und nicht nur die erreichten Punkte)

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 (47,99 Euro)

H. Klaeren, M. Sperber: Schreibe Dein Programm!
2014, <http://deinprogramm.de/>

M. Felleisen, R.B. Findler, M. Flatt, S. Krishnamurthi:
How to design programs, MIT, 2001 (80,00 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-WS14/

Kurzübersicht:

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