Geplante Gliederung der Vorlesung

Einführung in die Informatik III (3 V, 2 Ü)

im Wintersemester 2007/08

Dozent: Prof. Dr. Volker Claus, Universität Stuttgart

1. Einordnung und Inhalte

Sie haben in der Grundvorlesung Informatik den Rohstoff „Information“ und seine Darstellung, Verarbeitung, Speicherung und Übertragung sowie das Zusammenwirken von Prozessen kennen gelernt. Im Vordergrund stand meist eine „sequenzielle imperative Denkweise“, d. h., Variablen werden als Behälter mit erlaubten Werten, Operatoren und einem Verfallsdatum aufgefasst, auf denen Algorithmen schrittweise, einzeln und nacheinander mit nur einem Prozessor ausgeführt wurden. Datentypen können durch gängige Operationen zu sehr komplexen Gebilden zusammengesetzt werden, Verweise auf zugehörige Behälter, aber auch auf andere Programmeinheiten sind zulässig, die Zusammenfassung zu einem Modul (Paket), bestehend aus Spezifikation und Implementation, und deren Archivierung und Wiederverwendung erleichtern die Erstellung neuer und die Anpassung alter Programme. Hinzu kamen Komplexitäts- und Verifikationsfragen.

In der Veranstaltung „Einführung in die Informatik III“ geht es um die Vermittlung weiterer Denkweisen, um Probleme angemessen lösen zu können, konkret um
- objektorientiertes Programmieren,
- funktionales Programmieren und
- systemnahes Programmieren,
- (sofern Zeit bleibt:) Nebenläufigkeit.

Hierzu werden neue Sprachen und Kalküle eingeübt und zwar
- Java,
- Scheme,
- abstrakter Assembler, RAM,
- gemeinsame Variable bzw. Kommunikationskanäle.

Zugleich werden handwerkliche Fähigkeiten, speziell das Programmieren in Java und Scheme eingeübt. Daneben werden weitere Grundlagen der Informatik vermittelt, die in späteren Grundlagenveranstaltungen (Betriebssysteme, Übersetzer, Softwaretechnik usw.) aufgegriffen werden.

2. Rahmen der gesamten Veranstaltung

Die Veranstaltung dauert 16 Wochen (in der Zeit vom 16.10.07 bis 15.2.08). Sie besteht aus (1 „Vorlesungs- oder Übungsstunde“ entspricht 45 Zeit-Minuten):
Vorlesung: 3 Vorlesungsstunden pro Woche,
Übungen: 2 Übungsstunden pro Woche (ab zweiter Vorlesungswoche).

Zeit und Ort im Wintersemester 2007/2008:
Vorlesung: Dienstag und Donnerstag 14:00 bis 15:30 Uhr, Hörsaal 38.01.
Dozent: Prof. Dr. Volker Claus.
Übungen und Programmierübungen: Je nach Übungsgruppe.
Betreuung: Dipl.-Inf. S. Riexinger.

3. Anforderungen an Sie (beachten Sie: *Anwesenheit reicht nicht!*)

Sie sind nun im dritten Fachsemester. Wir erwarten, dass Sie fehlendes Wissen und fehlende Fertigkeiten selbst ausgleichen werden. Insbesondere sollten Sie Ihre Programmierfähigkeiten trainieren. Die Veranstaltung erfordert Ihre regelmäßige Mitarbeit, insbesondere das Einüben und Festigen des Stoffs der Vorlesung durch Nacharbeit und Übungen.
Der studentische Aufwand beträgt im Mittel für die Lehrveranstaltung wöchentlich 11 Zeitstunden, und zwar 2,25 Zeitstunden für die 3 Vorlesungsstunden, 1,5 Zeitstunden Übungen, 2,25 Zeitstunden Nacharbeit, 4 Zeitstunden für die Bearbeitung der Aufgaben und 1 Zeitstunde für die zusätzlichen Probleme mit dem Rechner. Wird diese Zeit regelmäßig (!) in der Vorlesungszeit aufgewendet, so reichen insgesamt weitere 30 bis 40 Stunden zur Prüfungsvorbereitung aus.
[Man rechnet überschlägig, dass jede(r) Studierende für eine Semesterwochenstunde (1 SWS) insgesamt rund 45 Zeitstunden aufzuwenden hat. Das heißt, dass Sie insgesamt 225 Stunden von der ersten Stunde bis zur Klausur investieren müssen; darin sind die 5 mal 16 mal 3/4 = 60 Zeitstunden, die Sie in den Lehrveranstaltungen anwesend sind, enthalten.]
Planen und überwachen Sie Ihren Zeitaufwand möglichst genau. Die Fortschrittsgeschwindigkeit der Vorlesung ist darauf abgestellt, dass Sie den Stoff der Grundvorlesungen in Informatik und in Mathematik sowie das Programmieren in Ada beherrschen. Der Stoff baut stets auf dem bereits Gelernten auf und so steigern sich einzelne Versäumnisse schnell zu großen, kaum noch zu überbrückenden Wissenslücken. Deshalb: Sollten Sie Probleme mit dem Verständnis haben oder können Sie nicht mehr als 50% der Übungsaufgaben alleine lösen, so sprechen Sie **sofort** mit dem Betreuer oder den Tutor(inn)en. Hierfür können z. B. E-Mails und die Sprechstunden und die direkte Ansprache in und nach den Lehrveranstaltungen genutzt werden.

4. Die Vorlesungsinhalte im Einzelnen

Die Vorlesung orientiert sich an der universitären Standardliteratur, siehe unten. Sie weicht inhaltlich von den Veranstaltungen der vergangenen Jahre ab, auch wenn die große Richtung und mindestens 60% des Stoffs übernommen werden.

Die Übungen werden sich bis Anfang Dezember 2007 vor allem auf das Problemlösen mit einer objektorientierten Sprache (hier: Java) konzentrieren. Ab dem 10.12. 2007 wird der funktionale Programmierstil eingeübt. Das konkrete Programmieren spielt eine zentrale Rolle, da die Konzepte sonst "nur graue Theorie" bleiben. Zugleich bereitet es auf das Softwarepraktikum in der Informatik und auf die Praxis in der Automatisierungstechnik vor.

Für die Verwaltung und Abgabe der Übungen wird das System eClaus eingesetzt.

Umfang der Vorlesung: 16 Wochen à 3 Vorlesungsstunden. Da keine Feiertage auf unsere Termine fallen, stehen 48 Vorlesungsstunden zu je 45 Zeit-Minuten im Wintersemester zur Verfügung, wobei eine Doppelstunde für Zwischenklausuren und Evaluationen einzuplanen ist. Wir unterteilen die 24 Doppelstunden in 4 Abschnitte und eine Vorbemerkung. Die Veranstaltungstermine und die geplanten Inhalte lauten:

Geplante Gliederung der Vorlesung Einführung in die Informatik III:

0. Vorbemerkungen	16.10.
0.1 Hinweise zu Unterlagen, Durchführung, Übungen, Programmieren, Ada.	
0.2 Formalismen (BNF, Syntax)	
0.3 Hinweise zur Softwareerstellung	
1. Objektorientierte Programmierung (11 Doppelstunden)	
1.1 Prinzipien (Klassen, Objekte, Methoden, Kommunikation, Vererbung)	22.10.
1.2 Eine konkrete oo-Sprache (Einführung in Java, Teil 1)	23.,29.,30.10.
1.3 Java, Teil 2	06. und 12.11.
1.4 Problemlösen, Softwareerstellung, Beispiele	13. und 19.11.
1.5 Java, Teil 3	26.11.
1.6 Beispiele, Weitere objektorientierte Sprachen	27.11.
(Ergänzungen / *Zwischenklausur*)	03.12.
2. Funktionale Programmierung (8 Doppelstunden)	
2.1 Prinzipien, Beispielprogramm	10.12.
2.2 Typen, Funktionen, Listen, elementare Sprachelemente in Scheme	11. und 18.12.
(Weitere Beispiele / *Lehrevaluation*)	08.01.
2.3 Höhere Ordnungen, Beispiele in Scheme	14. und 15.01.
2.4 Listen	21.01.
2.5 Ausblick und andere funktionale Sprachen (*und Zwischenklausur*)	22.01.
3. Systemnahe Programmierung (3 Doppelstunden)	
3.1 Registermaschine	28.01.
3.2 Stackmaschine	29.01.
3.3 Attributierte Grammatiken	11.02.
4. Nebenläufigkeit (sofern noch Zeit bleibt)	12.02.
(Stichwörter: Gemeinsame Variable, Kanäle, S/T-Netze, Farkas)	

Literatur:

Abelson, Harold, and Gerald J, Sussman with J. Sussman, “Structure and Interpretation of Computer Programms”, MIT, 1996 (es gibt auch eine deutsche Ausgabe, 3. Auflage, Springer, 1998)

Bishop, Judy, "Java lernen", Pearson-Studium, Addison-Wesley, 2. Auflage, 2003

Boles, D.: "Programmieren spielend gelernt mit dem Java-Hamster-Modell", Teubner-Verlag, Wiesbaden, 2006

Cousineau, Guy and Michel Mauny, „The Functional Approach to Programming“, Cambridge University Press, ISBN: 0521576814 (1998)

Goos, Gerhard und Wolf Zimmermann, „Vorlesungen über Informatik Band 1: Grundlagen und funktionales Programmieren“, Springer, ISBN: 3540244050 (2005)

Goos, Gerhard und Wolf Zimmermann, „Vorlesungen über Informatik Band 2: Objekt-orientiertes Programmieren und Algorithmen“, Springer, ISBN: 3540244034 (2006)

Heinisch, Cornelia, Müller-Hofmann, Frank, und Goll, Joachim, "Java als erste Programmiersprache", Teubner-Verlag, Wiesbaden, 2007

Loeckx, J., Mehlhorn, K., Wilhelm, R., „Grundlagen der Programmiersprachen“, Teubner-Verlag, Stuttgart 1986

Ottmann, T., und Widmayer, P., „Algorithmen und Datenstrukturen“, Spektrum Verlag, Heidelberg, 4. Auflage 2002

Scott, Michael L., „Programming Language Pragmatics“, Morgan Kaufmann Publishers, ISBN: 0126339511 (2000)

Thiermann, Peter, „Grundlagen der funktionalen Programmierung“, Teubner Verlag, ISBN: 3519021374 (1994)

Wilson, Paul, "An introduction to Scheme and its Implementation", online-Buch: ftp://ftp.cs.utexas.edu/pub/garbage/cs345/schintro-v14/schintro_toc.html

5. (Vordiploms-) Prüfungen und erforderlicher Schein

Für Informatik-Studierende heißt die Vordiplomsprüfung:
"Praktische Informatik II",
für Studierende der Automatisierungstechnik:
"Einführung in die Informatik III".

Die Prüfungen sind für die verschiedenen Studiengänge einheitlich. Es wird der Stoff der Vorlesung und der Übungen geprüft. Als Voraussetzung für die Teilnahme an der Vordiplomsklausur benötigen die Informatik-Studierenden den Übungsschein von „Einführung in die Informatik III“ oder den von „Einführung in die Informatik I und II“. Die Automatisierungstechniker(innen) benötigen den Übungsschein „Einführung in die Informatik I und II“.
Die für alle Studiengänge einheitliche schriftliche Prüfung findet voraussichtlich am Mittwoch, dem 5. März 2007 statt und dauert eine Stunde. Als Hilfsmittel sind zwei DIN A4 Blätter mit handschriftlichen Notizen zugelassen.
Die Prüfung ist beim Prüfungsamt anzumelden (in der Zeit vom 26. bis 30.11.07).
Es finden zu dieser Veranstaltung zwei Zwischenklausuren statt, deren Punktzahlen bei den Übungspunkten mitgezählt werden. Geplante Termine hierfür: 3.12.07 und 22.01.08.

6. Übungen zur Vorlesung "Einführung in die Informatik III"

Die für den Übungsschein erforderlichen Leistungen bestehen aus den wöchentlich abzugebenden Übungsaufgaben (maximal 13 mal 20 Punkte) und zwei Klausuren (zu je 30 Punkten) und mindestens 2 Präsentationen in den Übungsgruppen. Der Schein wird am Ende der Vorlesungszeit ausgestellt, wenn die Präsentationen erfolgreich waren, wenn in den letzten vier Übungsblättern mindestens 40 Punkte erreicht wurden und wenn mindestens 60% der maximal erreichbaren Punkte (also mindestens 192 Punkte) erreicht wurden.
Wie im ersten Studienjahr werden Übungsgruppen gebildet.
Zu weiteren Einzelheiten siehe Extrazettel „Organisatorisches“.

7. Organisieren Sie Ihre Arbeit!

Nun wissen Sie es ja schon: Sie müssen Ihre Arbeit einteilen, Ihre verfügbare Zeit gut planen und sich an Ihre eigenen Vorgaben auch halten! Entwickeln Sie Ihre eigenen Techniken, wie Sie den Stoff am besten lernen, weiter. *Überwachen Sie Ihre Arbeitszeit:* Schreiben Sie genau auf, wie viel Zeit Sie für was aufwenden und korrigieren Sie ggf. Ihre Gewohnheiten. In der Regel wird Ihr Intellekt durch „häufiges Sitzen am Rechner“ nicht geschult, weil Sie hierbei keine Konzepte und Lösungswege durchdenken. Diskutieren Sie den Stoff und die Aufgaben mit anderen, um die Inhalte zu festigen und selbst festzustellen, wo Sie stehen, was Sie gut oder nicht verstanden haben, wo Sie noch üben müssen usw. Werden Sie sich selbst und anderen gegenüber fachlich immer kritischer.

Verantwortliche Personen: (Name, Raum, Telefon: 0711 / 7816 - ..., E-Mail)
Prof. Dr. Volker Claus, Raum 1.111, -300, claus@informatik.uni-stuttgart.de
Dipl.-Inf. Sascha Riexinger, Raum 1.152, -337, riexinger@informatik.uni-stuttgart.de
Heike Photien (Sekr.), Raum 1.117, Tel.: - 328, photien@informatik.uni-stuttgart.de
Sprechstunde V. Claus: in der Regel Mi. 13-14 Uhr (bitte trotzdem anmelden).