Geplante Weiterführung der Grundvorlesung (4V, 2Ü, 2PÜ)

Einführung in die Informatik II

Sommersemester 2003

1. Überblick über die Einführung in die Informatik I

0. Vorbemerkungen (ca. eine Doppelstunde)
0.1 Zum Studium
Planung im Studium, Selbst-Überwachung, Arbeitsleistung, Arbeitsmoral, Ziele des Studiums, Ziele der Vorlesung, Voraussetzung für diese Vorlesung, Ablauf und Ihre Mitwirkung, Zusammensetzung der Vorlesung, Bücher, Skript, Folienkopien, Mitschrift, „elektronisches Textbuch“ , Unerwünschtes, Fragen?
0.2 Hinweise zu den Unterlagen, zur Durchführung, zu Übungen und Programmierkurs
0.3 Formalismen (Mathematische Grundlagen; diese werden in die einzelnen Kapitel integriert).

1. Grundlagen der Programmierung (21.10.02 – 13.2.03)

1.1 Algorithmen und Sprachen
1.1.1 Darstellung von Algorithmen (auch: Ablaufprotokolle)
1.1.2 Grundlegende Datenbereiche (ggT, ASCII, Komplementdarstellung, Fest-/Gleitpunkt, Rundungsfehler, Skalarprodukt)
1.1.3 Realisierte Abbildung (auch: Berechenbarkeit)
1.1.4 (Künstliche) Sprachen
1.1.5 Grammatiken (kontextfrei, erzeugte Sprache, Ableitung(sbaum), Klassifikation)
1.1.6 BNF, Syntaxdiagramme (auch EBNF)
1.1.7 Sprachen zur Beschreibung von Sprachen (EBNF für alle EBNFs)
1.1.8 Historische Anmerkungen
1.1.9. Übungsaufgaben

1.2 Aussagen über Algorithmen
1.2.1 Charakteristika von Algorithmen
1.2.2 Grenzen der Algorithmen, Unentscheidbarkeit (Halteproblem, ε-H.)
1.2.3 O-Notation
1.2.4 Darstellung durch Gleichungen (Teilworterkennung, Quersumme)
1.2.5 Historische Anmerkungen
1.2.6. Übungsaufgaben

1.3 Daten und ihre Strukturierung
1.3.1 Elementare Datentypen (auch Codes)
1.3.2 Konstruktoren (für Datenbereiche) (Beispiele: Speicherabbildungsfunktion, binäre Suche)
1.3.3 Relationen, Graphen, Referenzen (Listen, Keller, Schlange, deque)
1.3.4 Keller und Halde

1.4/1.5 Programmierung (und die Sprache Ada 95)
1.4.1 Blöcke, Deklarationen und Ausnahmen (Kellerspeicher, Sichtbarkeit)
1.4.2 Prozeduren und Funktionen (Rekursion, Kopierregel, Seiteneffekte, Operatoren, überladen)
1.4.3 Moduln (und Pakete)
1.4.4 Polymorphie (und Generizität)
1.4.5 Vererbung (einfach, mehrfach, abstrakte Klasse)
1.4.6 Abstrakte und konkrete Datentypen (Boolesche Werte, lbs)
1.4.7 Objekte (Zahlen, Geometrie, Skat)
1.6 Komplexität von Algorithmen und Programmen

1.6.1 Rechenmodell "Turingmaschine" (det., nichtdet.; wiederholungsfreie Wörter)
1.6.2 Churchsche These
1.6.3 Komplexitätsklassen (D/NTime, D/NSpace, P, NP)
1.6.4 Beispiele (ggT, Schnellere Multiplikation, Sortieren mit Bäumen, Graphen, Durchlauf von Bäumen und Graphen, Transitive Hülle eines Graphens, starke Zusammenhangskomponenten, Morse Code, Binpacking)
1.6.5 Andere Rechenmodelle (TM, LBA, PDA, Endl. Automaten, Nichtdeterminismus)
1.6.6 Historische Anmerkungen

Faktisch standen von den vorhandenen 31 Doppelstunden rund 28 Doppelstunden zur Verfügung; drei Doppelstunden wurden für Allgemeines, die Tests und Vorlesungsbefragungen verwandt. Die Lehrveranstaltung (rund 850 Folien) steht komplett im Netz. Es wurden 3 Tests durchgeführt, die annehmbar ausfielen. Die Zahl der Hörer(innen) nahm von 305 (=Anmeldungen zu den Übungen im Oktober 2002) bzw. 230 Personen nach drei Wochen auf rund 120 Anwesende in der letzten Vorlesungsstunde ab. Die Klausur am 20.2.03, die nur von wenigen Studierenden geschrieben werden musste, hat fast niemand bestanden. Die Übungsscheine in Informatik I und im Programmierkurs wurden von mehr als 50% der jeweiligen Teilnehmer(innen) erworben.

2. Plan der Vorlesung im SS 2003

1.7 Semantik von Programmen (28.4. und 5.5.03, Folien bereits im Netz)
Korrektheit, Zusicherungen, schwächste Vorbedingung,
Beispiele, Terminierung

2. Interaktionen

2.1. Objektorientierung (8.5.)
Klassen und Instanzen, Vererbung, Nachrichtenaustausch
(Ergänzungen zu 1.4.7; Darstellungen in Ada)

2.2. Prozesse (12.5., 15.5.)
Nebenläufigkeit, Parallelität, Ereignisse, S/T-Netze
Synchronisation, asynchrone Prozesse, Protokolle
Gemeinsame Variablenbereiche, PRAM
Realisierung in Ada

2.3. Vernetzte Systeme (19.5., 22.5.)
Verteilte Algorithmen, verteilte Daten
Client-Server-„Architektur“
Internet, Informationsdienste

2.4. Modellierung (sofern Zeit bleibt, 26.5.);
Testklausur 1

3. Grundlegende Verfahren (ab 2.6.)

3.1. Basisverfahren
Zufallszahlen, Permutationen, Entscheidungs-/Optimierungsprobleme, Entscheidungstabellen, Symboltabellen, Verwaltung von Kellern (Garwick-Algorithmus), Verwaltung der Halde (Dynamische Daten, Freispeicherlisten, Buddy, garbage collection)

3.2. Suchen
Suchen in flachen Strukturen
Wiederholung: Binäre Bäume, Suchbäume, Binarisierung von Bäumen,
Durchsuchen von Bäumen (Traversierung)
Balancierte Bäume, AVL-Bäume, B-Bäume
Digitale Suchbäume

3.3. Hashing
Überblick, Ideen, Beispiel "modulo p", Hashfunktionen
Offenes Hashing, Kollisionsverfahren (Sondierungen)
Rehashing
Testklausur 2

3.4. Sortieren
Überblick, allgemeine Aussagen
Aussuchen, Auswählen (Min., Heapsort mit Varianten)
Einfügen (Listen, Bäume, Radix, Shell)
Austauschen (Bubblesort, Quicksort)
Mischen (Mergesort und diverse Varianten)
Streuen und Sammeln (Bucketsort)
Paralleles Sortieren (odd-even-merge)
Vorlesungsumfrage der Universität

3.5. Graphalgorithmen
Überblick, Definitionen
Topologisches Sortieren
Kürzeste Wege
Minimale Spannbäume
Zusammenhangskomponenten
Flüsse in Netzen
Testklausur 3

3.6. Geometrische Algorithmen (sofern Zeit bleibt)
Z.B.: Schnittproblem für Rechtecke, Konvexe Hülle, Minimale Entfernungen
Eine Sprache für die Elementargeometrie, Realisierung in Ada

Klausur am 5.8.03 ab 8:30 Uhr, Pfaffenwaldring 47 und 53.

Rahmen: Die Veranstaltung besteht aus (1 „Stunde“ sind hier 45 Zeit-Minuten):
Vorlesung: 4 Vorlesungsstunden pro Woche,
Übungen: 2 Übungsstunden pro Woche,
Programmierübungen: 2 Übungsstunden je Woche.
Die Programmierübungen sind Pflicht für Studierende der Informatik und der Wirtschaftsinformatik. Allen anderen wird empfohlen, an den Programmierübungen teilzunehmen.

Hinweis: Die Erfahrungen der letzten Jahre sagen, dass etwa die Hälfte unserer Studierender eine eigenartige Arbeitsmethodik besitzen und nur das lernen, was minimal gefordert wird. Zum Bestehen der Vordiplomsprüfungen genügt dieses nicht. Beachten Sie, dass in der Klausur der Stoff von 160 Veranstaltungsstunden (108 Stunden Vorlesung, 52 Stunden Übungen) geprüft wird. Hier reichen der informative Besuch der Veranstaltungen und die anschließende Vorbereitung mit vielleicht 100 Zeitstunden in der Regel nicht aus. Daher müssen Sie im Sommersemester kontinuierlich mitarbeiten und zusätzlich die Tage vom letzten Vorlesungstermin bis zum 5.8. gezielt nutzen!

Für Personen, die weitere Beispiele kennen lernen möchten oder die den Stoff nochmals erläutert haben wollen, wird erneut ein „Zusatzkurs“ angeboten. Hauptsächlich für Nebenfachstudierende bieten wir zusätzlich auch ein „Kolloquium Programmierung“ an, in dem wir auf die Programmieraufgaben in der Klausur vorbereiten.

Die Teilnehmer(innen) sind aufgefordert, den Kontakt zu den wissenschaftlichen Mitarbeiter(inne)n und zu den Tutor(inn)en zu suchen. Hierfür können z.B. Emails und die Sprechstunden genutzt werden. Die Studierenden sollten einen oder zwei Vorlesungssprecher(innen) benennen, die allgemeine und spezielle Kritik mit dem Lehrpersonal besprechen und Maßnahmen zum besseren Erfolg der Veranstaltung vorschlagen sollen.

3. Literatur

Manuskripte von Prof. Plödereder, SS 01 und Ergänzungen von Prof. Claus (SS 02).
Appelrath, Hans-Jürgen und Ludewig, Jochen, "Skriptum Informatik - eine konventionelle Einführung", Verlag der Fachvereine Zürich und B.G. Teubner Stuttgart, 4. Auflage 1999
Balzert, Helmut, "Lehrbuch Grundlagen der Informatik", Spektrum Akademischer Verlag, Heidelberg 1999
Broy, Manfred, „Informatik. Eine grundlegende Einführung“. Band 1: Programmierung und Rechnerstrukturen, Springer-Verlag, 1998. Band 2: Systemstrukturen und Theoretische Informatik, Springer-Verlag, 1998
Cormen, Leiserson, Rivest, "Introduction to Algorithms", MIT Press, 1996 (zweite Auflage 2001)
Goos, Gerhard, "Vorlesungen über Informatik", Band 1 und 2, dritte Auflage, Springer-Verlag, Berlin 2000 und 2001
Gumm, H.-P., Sommer, M., „Einführung in die Informatik“, 5. Auflage, Oldenbourg-Verlag, München 2002
Güting, R.H., "Datenstrukturen und Algorithmen", B.G.Teubner Stuttgart, Neuauflage 2003
Klaeren, Herbert, Sperber, Michael, "Vom Problem zum Programm", 3. Auflage, B.G. Teubner Stuttgart, 2001
Loeckx, J., Mehlhorn, K., Wilhelm, R., "Grundlagen der Programmiersprachen", Teubner-Verlag, Stuttgart 1986
Ottmann, T., Widmayer, P., "Algorithmen und Datenstrukturen", Spektrum Verlag, Heidelberg, 1996
Schöning, Uwe, "Algorithmik", Spektrum Akademischer Verlag, Heidelberg 2001
Sedgewick, Robert, "Algorithms in C", 3rd Edition, Addison-Wesley, 1998
sowie "Algorithms in Java", Addison-Wesley, 1998

Bücher zur Sprache Ada 95:
Barnes, J.G.P., „Programming in Ada 95“, 2. Auflage, Addison-Wesley 1998
Barnes, John, „Ada 95 Rationale. The Language - The Standard Libraries”, Springer-Verlag, Lecture Notes in Computer Science (Vol. 1247), 1997
Feldman, M.B. und Koffman, E.B., „Ada 95 - Problem Solving and Program Design“, 2. Auflage, Addison-Wesley 1997 (3. Auflage Textbook Binding, 1999)
Nagl, M., „Softwaretechnik mit Ada 95. Entwicklung großer Systeme“, Vieweg-Verlag, Wiesbaden 1999

Weitere Literatur
Für diverse Definitionen und Erläuterungen:
"Duden Informatik", dritte Auflage, Bibliografisches Institut, Mannheim, 2001.
Für Mathematik und Ideen:
Kiyek, K. und Schwarz, F., „Mathematik für Informatiker 1 und 2, Teubner-Verlag, 3. bzw. 2. Auflage, 2000 und 1999
Meinel, Christoph und Mundhenk, Martin, „Mathematische Grundlagen der Informatik“, Teubner-Verlag, Wiesbaden, 2. Auflage, 2002.
Schöning, Uwe, „Ideen der Informatik“, Oldenbourg-Verlag, München, 2002.

4. (Vordiploms-) Prüfungen

Die Prüfung findet am 5.8.03 als Klausur statt. Die Teilnahme muss beim Prüfungsamt angemeldet werden. Voraussetzung ist in der Regel mindestens ein Schein.
Über Informatik I und Informatik II finden eigene Klausuren ebenfalls am 5.8. statt.
Derzeit gibt es folgende Varianten betreffend die Vorlesungen „Einführung in die Informatik I, II, III“ für die verschiedenen Studiengänge:

1a. Orientierungsprüfung für den Diplomstudiengang Informatik, Prüfung Praktische Informatik I (1. Teil über Informatik I, II im Vordiplom): 2 Stunden Klausur.
1b. Praktische Informatik II (2. Teil über Informatik III, Vordiplom Informatik): 1 Stunde.
2. Alte Prüfungsordnung Informatik: Praktische Informatik (über die drei Vorlesungen Informatik I, II, III): 3 Stunden.
3. Orientierungsprüfung für den Diplomstudiengang Softwaretechnik, Prüfung Praktische Informatik A (Informatik I und II), 2 Stunden Klausur
4a. Einführung in die Informatik I und II, Vordiplom Automatisierungstechnik: 2 Stunden.
4b. Einführung in die Informatik III, Vordiplom Automatisierungstechnik: 1 Stunde.
5. Einführung in die Informatik I und II, Vordiplom Computerlinguistik: 4 Stunden.
6. Informatik (Informatik I und II) für Geoinformatik: 4 Stunden.
7a. Einführung in die Informatik I, Betriebswirtschaftslehre: 2 Stunden.
7b. Einführung in die Informatik II, Betriebswirtschaftslehre: 2 Stunden.
7c. Einführung in die Informatik III, Betriebswirtschaftslehre: 1 Stunde.
8a. Informatik (Informatik I und II), Nebenfach zur Mathematik: 4 Stunden.
8b. Einführung in die Informatik III, Nebenfach zur Mathematik: 1 Stunde.
9a. Einführung in die Informatik I für den Bachelorstudiengang Wirtschaftsinformatik: 2 Stunden Prüfung (Prüfungsleistung), kann für die Orientierungsprüfung verwendet werden.
9b. Einführung in die Informatik II für den Bachelorstudiengang Wirtschaftsinformatik: 1 Stunde Prüfung (Prüfungsleistung, für die Orientierungsprüfung anrechenbar).
10. Wahlfachprüfung eventuell (Physik, Technische Kybernetik, Magisterstudiengänge...): zwei bis vier Stunden Prüfung.

Die meisten Prüfungsordnungen sehen vor, dass eine nicht bestandene Prüfung beim nächsten Prüfungstermin (oder innerhalb eines halben Jahres) wiederholt werden muss. In der Regel kann man die Prüfung nur einmal wiederholen, doch eventuell gibt es in Ihrer Prüfungsordnung eine „Freiversuch“-Regelung oder eine „mündliche Nachprüfung im unmittelbaren zeitlichen Zusammenhang nach der Bekanntgabe des zweiten Prüfungsergebnisses“. Prüfen Sie dies selbst nach.

Benotung der Prüfung
In allen Studiengängen ist die Note in dem jeweiligen Fach identisch mit der Ergebnisnote in der Klausur.

5. Übungen zur Vorlesung

Parallel zur Vorlesung werden wöchentlich Übungen angeboten, die unter Leitung eines Mitarbeiters oder Tutors in Gruppen durchgeführt werden. Jede Gruppe besteht aus 18 bis 20 Personen. Es wird dringend empfohlen, an diesen Übungsgruppen aktiv teilzunehmen. Es werden voraussichtlich 12 Gruppen eingerichtet.

Wissenschaftliche Mitarbeiter (in Klammern die Zahl von ihnen betreuten Übungsgruppen) im Sommersemester: J. Holub (2), S. Lewandowski (2), W. Schmid (1), S. Zimmer (1).
Tutoren im Sommersemester: P. Jehlicka (2), H. Kahl (2), O. Schiller (2).

Termine: Montag 09:45 bis 11:15 Uhr,
Dienstag 09:45 bis 11:15 Uhr, 11:30 bis 13:00 Uhr, 15:45 bis 17:15 Uhr,
Mittwoch 09:45 bis 11:15 Uhr, 14:00 bis 15:30 Uhr,
Donnerstag 14:00 bis 15:30 Uhr

Es gibt im Sommersemester 12 Übungsblätter, das letzte Übungsblatt wird nicht mehr besprochen und bewertet.

Eintragen in die Übungsgruppen: Dies erfolgt online unter
https://inf2.informatik.uni-stuttgart.de/uebungsgruppen-bin/inf2/groups
Der Benutzername lautet "inf2", das Passwort "baum". Achten Sie darauf, Ihre Daten korrekt einzugeben. Insbesondere muss die Email-Adresse gültig sein.

Ablauf: Die Übungsblätter mit 2 bis 3 schriftlich abzugebenden Aufgaben, mit 2 bis 3 Votier-Aufgaben und eventuell mit einer Zusatzaufgabe liegen ab Freitag unter
http://www.informatik.uni-stuttgart.de/fmi/fk/lehre/ss03/info_II/default.htm
im Netz, erstmals am 25.4.03.
Die Ausgabe der Übungsblätter in Papierform erfolgt in der folgenden Montag-Vorlesung.

Die Aufgaben werden so gestellt, dass ihre vollständige Bearbeitung höchstens 5 Zeitstunden für einen „durchschnittlichen Studierenden“ beansprucht. Wer sich weiter oder in einer anderen Richtung vertiefen will, sollte mögliche Zusatzaufgaben bearbeiten.

Sofern Unklarheiten: Sprechstunde der Übungsgruppenleiter beachten. Erläuterungen werden ggf. ins Netz gestellt. Fragen können auch per Email gestellt werden und werden, falls von der Kapazität aus gesehen möglich, bis zum Ende des nächsten Arbeitstages beantwortet.
Fragen sowie Anregungen und Kritik können auch auf dem Schwarzen Brett
http://fachschaft.informatik.uni-stuttgart.de/forum
diskutiert werden.

Abgabe der Lösungen: bis zum nächsten Freitag 20:00 Uhr elektronisch, d.h. Bearbeitungszeit nach Ausgabe: 7 Tage.
Zur Bearbeitung und Abgabe wird das System eClaus verwendet. Eine kurze Einführung ist auch auf der Vorlesungsseite (siehe oben) zu finden.

Die Übungsaufgaben werden in den Übungsgruppen am der Abgabe folgenden Termin besprochen. Hierbei tragen die Bearbeiter(innen) ihre Lösungen vor.
Die korrigierten Abgaben sind spätestens ab der darauf folgenden Übungsstunde im eClaus System einzusehen.
Lösungshinweise oder sogar Musterlösungen werden einige Tage nach dem letzten zugehörigen Termin einer Übungsgruppe ins Netz gestellt. Zugleich sollen Auswertungen (Anzahl der Bearbeitungen, Durchschnittspunktzahlen der bearbeiteten Aufgaben, Hinweise auf Qualität der Abgaben) veröffentlicht werden.

Gruppenabgaben

Es ist zugelassen, dass bis zu 3 Personen gemeinsam eine Lösung abgeben. Kann ein Student eine von seiner Gruppe bearbeitete Lösung in der Übungsgruppe nicht vortragen, so werden ihm für dieses Übungsblatt sämtliche Punkte aberkannt. Im Wiederholungsfall werden *alle bis dahin erreichten Punkte* gestrichen.
Liegen bei mehreren Gruppen offensichtlich untereinander kopierte Lösungen vor, werden die Punkte der betroffenen Abgabegruppen halbiert.

Vortrag

Jedes Mitglied einer Übungsgruppe muss mindestens drei Mal im Semester in der Gruppe einen Vortrag halten, in dem jeweils die Lösung einer Übungsaufgabe vorgestellt wird.

Test-Klausuren während der Lehrveranstaltung

Es finden im Sommersemester an drei Terminen (voraussichtlich am 26.5., 26.6. und 21.7.) Testklausuren im Umfang von je 30 Minuten Dauer im Rahmen der Vorlesung statt. Die Klausuren dienen zum einen dem Selbsttest, zum anderen gehen sie in die Berechnung der Punktzahl für den Erwerb des Scheins ein. Daher sollte an diesen drei Klausuren jede(r) unter Angabe ihres/seines Namens (also nicht anonym) teilnehmen.

Punktzahlen und Schein
In jedem Übungsblatt können 20 Punkte erreicht werden. Ein Punkt entspricht rund 15 Minuten Bearbeitungszeit. In jeder Testklausur können 30 Punkte erreicht werden. Maximal können somit 310 Punkte im Sommersemester erreicht werden. Eine sehr gute Leistung liegt vor, wenn mindestens 279 Punkte erzielt wurden.
Sofern mindestens 155 Punkte (davon mindestens 45 Punkte in den Testklausuren) erreicht werden, der/die Betreffende mindestens drei Aufgaben korrekt vorgetragen hat und durch die Beteiligung in den Übungsgruppen oder durch andere Indizien ersichtlich ist, dass die Übungsaufgaben selbst bearbeitet und die Lösungen verstanden wurden, wird ein Schein über die erfolgreiche Teilnahme ausgestellt. Dieser Schein ist benotet und enthält die erreichte Punktzahl; die Note lautet:

4,0	ab 50% der erreichbaren Punkte
3,7	ab 55% der erreichbaren Punkte
3,3	ab 60% der erreichbaren Punkte
3,0	ab 65% der erreichbaren Punkte
2,7	ab 70% der erreichbaren Punkte
2,3	ab 75% der erreichbaren Punkte
2,0	ab 80% der erreichbaren Punkte
1,7	ab 85% der erreichbaren Punkte
1,3	ab 90% der erreichbaren Punkte
1,0	ab 95% der erreichbaren Punkte

Auf Wunsch werden die Angaben der Note und/oder der Punktzahl weggelassen.

6. Programmierkurs, Programmierübungen

Wissenschaftlicher Mitarbeiter im Sommersemester: S. Schwoon (2).
Tutoren für die Übungen im SS: M. Holz (2), J. Kizler (2), F. Schlachter (2).

Diese Veranstaltung ist nur für die Studierenden der Informatik und der Wirtschaftsinformatik verpflichtend. Sie besteht fast ausschließlich aus Übungen. Der Übungsbetrieb läuft wöchentlich analog zu den Übungen zur Grundvorlesung ab. Allerdings fallen zusätzlich gerätetechnische/softwaretechnische Fragen für die Programmierung in Ada 95 an. In der Regel sind jede Woche drei Programmieraufgaben zu bearbeiten, wobei jeweils 20 Punkte erworben werden können. Insgesamt werden 11 Übungsblätter ausgegeben, so dass 220 Punkte erreicht werden können. Die Regelungen von Abschnitt 5 gelten analog für die Ausstellung dieses Scheins (aber die Testklausuren entfallen). Studierende der Informatik und Wirtschaftsinformatik müssen diesen Schein über die erfolgreiche Teilnahme erwerben. Der Kurs ist aber auch für weitere Studierende offen.

Beginn und Termine:
In der ersten Woche wird das erste Übungsblatt ausgegeben. Der Übungsbetrieb beginnt in der zweiten Woche. Die genauen Termine finden Sie auf folgender Seite:
http://www.fmi.uni-stuttgart.de/szs/lehre/ss2003/ada95/index.shtml

Eintragen in die Übungsgruppen: Es erfolgt online unter
https://inf2.informatik.uni-stuttgart.de/uebungsgruppen-bin/infpk/groups
Der Benutzername lautet "infpk", das Passwort "baum". Achten Sie darauf, Ihre Daten korrekt einzugeben. Insbesondere muss die Email-Adresse gültig sein.

Inhalte des Sommersemesters: Ausnahmen, generische Pakete, Klassenkonzept in Ada, Nebenläufigkeit (tasks), größere Beispiele (über mehrere Übungsstunden) und Programmierprinzipien, ein wenig Software-Engineering und Richtlinien.