

Wichtiges Dokument - genau lesen und aufbewahren!

Geplante Gliederung der Grundvorlesung **Einführung in die Informatik I und II (4 V, 2 Ü)** **sowie des Programmierkurses I und II (2 V/Ü)**

im Studienjahr 1.10.05 bis 30.9.06

Zuständiger Dozent: Prof. Dr. Volker Claus, Universität Stuttgart

1. Zur Informatik

Informatik befasst sich mit dem Rohstoff „Information“ und seiner Darstellung, Verarbeitung, Speicherung und Übertragung, insbesondere aus technischer Sicht. Da das menschliche Zusammenleben durch Informationen gesteuert wird, hat die Informatik nachhaltigen Einfluss auf unsere Gesellschaft, auf das Arbeitsleben und auf das Privatleben jedes einzelnen. Stichwörter sind die Automatisierung geistiger Abläufe, die Dienstleistungs- und die Wissensgesellschaft, die Beherrschbarkeit von Prozessen und die Methoden zur Bearbeitung unüberschaubarer Vorgänge. Der zielorientierte Umgang mit der Informationsverarbeitung gehört heute zur erfolgreichen Ausübung sehr vieler Berufe. Zum einen sichert diese Tatsache den meisten Universitätsabsolvent(inn)en der Informatik ihren Arbeitsplatz, zum anderen droht hier eine geistige Spaltung der Bevölkerung und des Arbeitsmarktes. Der zugehörige Begriff "digital divide" besagt bereits, dass die Objekte und Produkte der Informatik "diskret" (gut untereinander unterscheidbar, zeichenartig, digital) sind; sie sind zugleich abstrakt und nicht „anfassbar“ (d.h., für die menschlichen Sinne nicht unmittelbar erfahrbar) und Informatikprodukte erkennt man nur an den Auswirkungen.

Informatik ist heute überwiegend eine Ingenieurwissenschaft. Sie umfasst

- Grundlagen,
- ingenieurmäßige Vorgehensweisen und
- experimentelle Anteile.

Im ersten Studienjahr werden die Grundlagen, insbesondere Prinzipien und Konzepte, sowie mehrere ingenieurmäßige Vorgehensweisen, vor allem konstruktive Verfahren und ihre Umsetzung in eine konkrete Programmiersprache, betont.

Für das Verständnis der Informatik sind die *Abstraktion* und die *Formalisierung* besonders wichtig. Abstraktion bezeichnet das Herausarbeiten des Wesentlichen, wobei man sich von vielen konkreten Details befreit; unter Abstraktion versteht man auch den Übergang zu einer Metaebene; sie umfasst das Erarbeiten gemeinsamer Ideen oder Prinzipien aus den unterschiedlichsten (Anwendungs-) Gebieten, z.B. die einheitliche Darstellung von Strukturen in Programmiersprachen, von syntaktischen Gebilden bei natürlichen Sprachen oder von Rechenbäumen mit Hilfe von (kontextfreien) Grammatiken. Abstraktes Denken ist eine wichtige Voraussetzung für eine adäquate *Modellierung*, welche die Dinge der realen Welt in eine Modellierungssprache überträgt, aus der man die Modelle dann weiter in Programmiersprachen abbildet. Um diese Modelle maschinell verarbeiten zu können, müssen sie formalisiert werden. Formalisierung beschreibt den Prozess, Geschehnisse der realen Welt oder (unscharfe) Informationen zu präzisieren und zugleich in Kalkülen zu notieren.

2. Inhaltliche Ziele der Veranstaltung

Ziele der Veranstaltung sind die Vermittlung grundlegender Darstellungen und Prinzipien der Informatik, insbesondere

- Formalismen und Beschreibungssprachen, Modellbildung,
- Algorithmen und ihre Eigenschaften (Zeitkomplexität, Verifikation, Terminierung),
- Datenstrukturen und die Effizienz zugehöriger Verfahren,
- Prozesse und Netze,
- Grundbegriffe und Denkweisen der Programmierung,
- Abstraktion und Theorie, auf denen diese Begriffe basieren,
- Vorgehensweisen zur Problemlösung und die Realisierung und Implementierung der Konzepte,
- Beherrschung wichtiger Standardalgorithmen,
- genaue Kenntnis typischer Beispiele und ihrer Programmierung,
- genaue Kenntnis einer Programmiersprache (hier: Ada 95).

Die Grundvorlesung ist von zentraler Bedeutung für die gesamte Informatik, da sie die Basisstrukturen und Konstruktionsprinzipien (heutiger) informationsverarbeitender Systeme behandelt, wichtige Denkweisen zur Entwicklung von Verfahren vorstellt, langlebige Ideen und Methoden vermittelt und Beurteilungskriterien über Vor- und Nachteile von Prozessen und allgemein von Problemlösungen diskutiert.

Für die Studiengänge Informatik und Wirtschaftsinformatik wird die Vorlesung ergänzt durch einen Programmierkurs, in dem in diesem Jahr die Programmiersprache Ada 95 "wie ein Handwerk" gelehrt wird. Auch allen anderen Teilnehmer(inne)n der Grundvorlesung wird dringend empfohlen, sich an dem Programmierkurs in irgendeiner Form zu beteiligen. Hier werden die in der Grundvorlesung behandelten Konzepte konkret an diversen Beispielprogrammen eingeübt.

Zur Bedeutung der Programmiersprache: Bundesweit wurde in den letzten Jahren an den deutschen Hochschulen festgestellt, dass eine hohe Korrelation zwischen der tatsächlichen Beherrschung einer Programmiersprache und dem Verständnis von Informatikkonzepten besteht. Wer meint, dass praktische Programmierung "nur" ein Anhang zur Vorlesung sei, "hat schon verloren". Die Beschäftigung mit der Informatik bedarf stets der Untermauerung durch konkrete Programmierung, d.h., die Prinzipien und das konkrete Handwerk gehören in der Informatik zusammen.

3. Struktur der gesamten Veranstaltung

Die Veranstaltung dauert 16 Wochen (in der Zeit vom 17.10.05 bis 17.2.06). Sie besteht aus (1 „Vorlesungs- oder Übungsstunde“ entspricht 45 Zeit-Minuten):

Vorlesung: 4 Vorlesungsstunden pro Woche (17.10.05 bis 16.2.06), (4 SWS)
Übungen: 2 Übungsstunden pro Woche (ab 20.10.). (2 SWS)

Hinzu kommt die praktische Programmierung im Umfang von 2 SWS:

Programmierkurs: 0 bis 0,5 Vorlesungsstunden je Woche,

Programmierübungen: 1,5 bis 2 Übungsstunden je Woche.

Programmierkurs/-übungen sind Pflichtveranstaltungen für Studierende der Informatik und der Wirtschaftsinformatik. Studierenden der anderen Studiengänge wird empfohlen, diesen Teil der Veranstaltung ebenfalls zu besuchen.

Für Personen, die weitere Beispiele kennen lernen möchten oder die den Stoff nochmals erläutern wollen, wird ein „Zusatzkurs“ angeboten.

Zeiten und Ort im Wintersemester 2005/2006:

Vorlesung: Montag, Donnerstag 15:45 bis 17:15 Uhr, Hörsaal 38.01.

Dozent: Prof. Dr. Volker Claus.

Programmierkurs: Donnerstag 8:00 bis 9:30 Uhr (nicht regelmäßig), Hörsaal 38.01.

Dozent: Dipl.-Inf. Sascha Riexinger.

Zusatzkurs: Dienstag 8:00 bis 9:30 Uhr, Hörsaal 38.01.

Dozent: Dr. Wolfgang Schmid.

Übungen und Programmierübungen: Je nach Übungsgruppe.

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

Wir erwarten von Ihnen regelmäßige Mitarbeit, insbesondere das Einüben und Festigen des Stoffs der Vorlesung durch Nacharbeit und Übungen. Der studentische Aufwand beträgt für die gesamte Lehrveranstaltung (4 Stunden Vorlesung, 2 Stunden Übungen und 2 Stunden Programmierkurs/-übungen pro Woche) wöchentlich 17 bis 19 Zeitstunden für Studierende der Informatik und Wirtschaftsinformatik bzw. 12 bis 14 Zeitstunden für alle anderen Hörer(innen). Im Einzelnen:

- 4,5 Zeitstunden für den Besuch der Vorlesung und der Übungsgruppe,
- 3,5 bis 5 Stunden Nachbereitung des Vorlesungsstoffs (individuell oder in einer Gruppe)
- 3 bis 5 Stunden Bearbeitungsdauer der Übungsaufgaben der Grundvorlesung in Gruppenarbeit,
- 1,5 Zeitstunden für die Teilnahme an Programmierkurs und -übungen,
- 3 bis 4 Stunden Bearbeitungsdauer der praktischen Programmieraufgaben.

Wird diese Zeit regelmäßig (!) in der Vorlesungszeit aufgewendet, so reichen insgesamt weitere 60 Stunden zur Prüfungsvorbereitung aus.

Man rechnet überschlägig, dass jede(r) Studierende für eine Semesterwochenstunde (1 SWS) insgesamt rund 45 Zeitstunden aufzuwenden hat. (Für die Bachelorstudiengänge: Ein Kreditpunkt entspricht rund 30 Zeitstunden.)

Sehr wichtiger Hinweis: Machen Sie sich einen Plan, wie Sie Ihre Arbeitszeit über die Woche auf Ihre Studienfächer verteilen und halten Sie diesen auch ein, vgl. Punkt 10 unten.

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. E-Mails und die Sprechstunden und die direkte Ansprache in und nach den Lehrveranstaltungen genutzt werden.

Wir bemühen uns darum, alle Hörer(innen) auf eine erfolgreiche Prüfung vorzubereiten. Defizite zeigen sich aber meist schon frühzeitig, vor allem beim Abschneiden in den Übungen und in den Zwischenklausuren. Leider haben wir nicht genügend Personal, um auf jede(n) einzelne(n) einzugehen. Lassen Sie Ihr Studium nicht schleifen, sondern *überwachen Sie sich selbst* und gehen Sie **sofort** in die Sprechstunden der Mitarbeiter oder des Dozenten, sobald Sie Ihre Leistungen nicht mehr über den Minimalanforderungen liegen. Die größte Gefahr, im Informatikstudium zu scheitern, liegt im schleichenden Ausklinken aus den Veranstaltungen! Denn der Stoff unserer Veranstaltung baut stets auf dem bereits Gelernten auf und daher steigern sich einzelne Versäumnisse schnell zu großen Wissenslücken.

Durch die Arbeit in Vierer-Gruppen und der anschließenden Kontrolle in den Übungsgruppen soll unter anderem auch diesem schleichenden Ausklinken entgegen gewirkt werden.

5. Voraussetzungen (Wissen/Können aus der Zeit vor dem Studium)

Vorausgesetzt werden mathematische Grundkenntnisse aus einer mindestens 4-SWS-Mathematikveranstaltung. Solche Veranstaltungen laufen parallel in den jeweiligen Studiengängen; das Wichtigste wird in der Vorlesung wiederholt. Programmierkenntnisse werden nicht vorausgesetzt. Solche Kenntnisse mögen hilfreich sein, sofern auf bereits Bekanntes zurückgegriffen wird, sie können aber auch gefährlich sein, wenn die Veranstaltung sich in neue Gebiete begibt und die Betroffenen diesen Wechsel nicht rasch genug realisieren. (In der Vorlesung wird für die Algorithmen meist eine vereinfachte Ada-ähnliche Notation verwendet, die auch Basis für Semantik, Verifikation und Komplexität ist. Viele Verfahren werden direkt in Ada 95 ausformuliert.)

Wer keine Programmierkenntnisse besitzt, ist daher keineswegs benachteiligt; er/sie muss sich jedoch von Anfang an möglichst gründlich in die Kontroll- und Datenstrukturen der Programmiersprache einarbeiten und Programme *alleine* zum Laufen bringen. (Lassen Sie sich hier auf keinen Fall von sog. "erfahrenen" Mitstudierenden helfen, sondern schreiben Sie Ihre Programme selbst!)

Weiterhin setzen wir voraus, dass Sie in der Vorlesungszeit mindestens 42, möglichst 45 Stunden pro Woche für Ihr Studium arbeiten. Durch die Weihnachts- und Pfingstpause und die Tatsache, dass die Prüfungen über die vorlesungsfreie Zeit gestreut werden, bleibt Ihnen hinreichend viel Zeit zum Wiederholen und zur Prüfungsvorbereitung. Nochmals die dringende Empfehlung: Teilen Sie Ihre Zeit gut ein und überwachen Sie dies!

Weiterhin setzen wir eine Grundmotivation bei Ihnen voraus. Wir werden uns nicht hinstellen und Sie von der Sinnhaftigkeit und den Vorteilen der Informatik überzeugen. Sie können gerne hierzu Fragen stellen, die dann in den Veranstaltungen aufgegriffen werden, aber solche Anstöße müssen von Ihnen kommen. Hierzu können Sie auch die Sprechstunden nutzen, da sich Fragen der Motivation und der Studienplanung viel besser in einem persönlichen Gespräch klären lassen.

6. Die Vorlesungsinhalte im Einzelnen

Die Vorlesung orientiert sich an der universitären Standardliteratur, siehe unten. Sie folgt inhaltlich den Veranstaltungen der vergangenen Jahre, die als Skript im Netz oder auf Papier bei der Fachschaft zur Verfügung stehen. Wie im jährlichen Rhythmus üblich werden Teile umgestellt, gewisse Gebiete anders betont und neue Inhalte auf Kosten älteren Stoffs hinzugefügt. Gegenüber früher werden im laufenden Jahr Aspekte der praktischen Programmierung verstärkt. Weiter wird viel Wert auf präzise Formulierungen gelegt, ohne die letztlich keine verlässlichen Programme geschrieben werden können.

Der Einstieg in die Informatik erfolgt dieses Mal über 6 Doppelstunden "Einführung in das Programmieren in Ada". Dies wird so gestaltet, dass reine Anfänger und Programmierfreaks gleichermaßen in neuen Stoff eingeführt werden.

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

Umfang der Vorlesung: 29 Wochen à 4 Vorlesungsstunden. Davon fallen 6 Stunden im Sommer 06 feiertagsbedingt und ca. 6 wegen der sechs Zwischenklausuren und der Lehrevaluationen aus. Zur Verfügung stehen somit 104 Vorlesungsstunden zu je 45 Zeit-Minuten. Wir unterteilen diese in 3 Abschnitte und eine Vorbemerkung.

Geplante Gliederung der Grundvorlesung:

0. Vorbemerkungen	17.10.
0.1 Hinweise zu den Unterlagen, zur Durchführung, zu Übungen und Programmieren	
0.2 Formalismen (<i>diese Teile werden nach Bedarf in die Vorlesung integriert!</i>)	
1. Kurze Einführung in die imperative Programmierung mit Ada	
Beispiele und Muster zum Wiederverwenden	17.10.
Begriffe Identifikator, Name, Variable; Ein- und Ausgabe; Files	18.10.
Syntaxbeschreibung, BNF	20.10.
Kontrollstrukturen (;, if, for, while, Block), Sichtbarkeit	20.10.
Funktion, rekursive Funktionen, Prozedur	27.10.
Datentypen: skalar, Felder, records, Untertyp, allgemeiner Begriff	03.11.
Einschub: Hinweise zum Informatik-Studium	07.11.
Planung im Studium, Selbst-Überwachung, Arbeitsleistung, Arbeitseinstellung	
Ziele des Studiums, Ziele der Vorlesung, Voraussetzungen für diese Vorlesung	
Ablauf und Ihre Mitwirkung, Unerwünschtes und Fragen	
Zusammensetzung der Vorlesung, Bücher, Skript, Folienkopien, Mitschrift	
2. Algorithmen und Sprachen	07.11.
2.1 Darstellung von Algorithmen, grundlegende Sprachelemente	07.11.
Grundlegende Datenbereiche, Realisierte Abbildung	
2.2 Künstliche Sprachen	10.11.
Regeln, Grammatiken, Sprachen (auch: BNF, Syntaxdiagramm)	
Sprachen zur Beschreibung von Sprachen, speziell von Algorithmen	
2.3 Algorithmen mit sehr kleinem Befehlssatz	17.11.
2.4 Charakteristika von Algorithmen (Aufwand und Bedeutung)	21.11.
Grenzen der Algorithmen, Unentscheidbarkeit, Churchsches These	
Berechnete Funktion (determiniert, deterministisch usw.)	
Komplexität von Algorithmen, O-Notation	24.11.
2.5 Darstellung durch Funktionen und Gleichungen	24.11.
Rekursion und Iteration	28.11.
Funktionaler Programmierstil	
3. Daten und ihre Strukturierung	
3.1 Elementare Daten	01.12.
Darstellungen (Zahlen, Zeichen, ...)	
3.2 Datentypen	05.12.
Konstruktoren (x, u, Wortmenge, Potenzmenge, Funktionenmenge)	
3.3 Relationen, Graphen, Referenzen (gerichtete Kanten)	08.12.
3.4 Listen, Keller, Halde	15.12.
4. Begriffe der Programmierung (und ihre Realisierung in Ada)	
4.1 Elemente von Programmen (Variablen, Dekl., sequentiell, verteilt, parallel, Stile)	15.12.
Elementare Datenbereiche, elementare Anweisungen	
Daten-Konstruktoren, Anweisungs-Konstruktoren	
4.2 Umgebungen (Blöcke, lokal, global, Gültigkeit, Lebensdauer)	22.01.
4.3 Deklaration von Algorithmen (Prozeduren, Funktionen)	
speziell: Parameterübergabe und Kopierregel	
4.4 Module, Pakete	09.01.
4.5 Polymorphie, Generizität, Relationen auf Datentypen (z.B. Vererbung)	12.01.
5. Abstrakte Datentypen	
5.1 Abstrakter und konkreter Datentyp, Instanz	16.01.
5.2 Objekte und Klassen	19.01.
Darstellung in Ada	
5.3 Objektorientierter Programmierstil	23.01.
6. Komplexität von Algorithmen und Programmen	
Elementares Rechenmodell (hier: Turing-Maschinen)	26.01.
Präzisierung der Komplexität, Komplexitätsklassen	30.01.
Andere Rechenmodelle	
Beispiel Backtracking	02.02.

Automatenmodell	06.02.
Beispiel Greedy	06.02.
Entwurfmethoden für Algorithmen	09.02.
7. Formale Semantik von Programmen	
7.1 Zusicherungen, beweisbare Korrektheit von Programmen	13.02.
Hoaresche Regeln und Beweisverfahren	
7.2 Schwächste Vorbedingungen	16.02.
7.3 Terminierung	
<i>-- Ende der Vorlesungszeit des Wintersemesters -- Wirtschaftsinformatik-Klausur: Mittwoch 8.3.06</i>	
8. Suchen	24.04.
8.1 Abstrakte Operationen, Suchen in sequentiellen Strukturen	
8.2 Bäume und (binäre) Suchbäume, Traversieren, Anzahlen	27.04.
8.3 Balancierte Bäume (insbesondere AVL-Bäume)	04.05.
8.4 B-Bäume	08.05.
8.5 Digitale Suchbäume (Tries)	11.05.
8.6 Datenstrukturen mit Historie	
9. Hashing	18.05.
9.1 Beispiel "modulo p"	
9.2 Hashfunktionen	
9.3 Offenes Hashing (und Sondierungen)	22.05.
9.4 Analyse der Hashverfahren	29.05.
9.5 Rehashing	01.06.
10. Sortieren	12.06.
10.1 Überblick, allgemeine Aussagen	
10.2 Aussuchen, Auswählen (Minimum, Heapsort)	19.06.
10.3 Einfügen (Listen, Bäume, Radix)	22.06.
10.4 Austauschen (Bubblesort, Quicksort)	26.06.
10.5 Mischen (Mergesort und diverse Varianten, externes Sortieren)	29.06.
10.6 Streuen und Sammeln (Bucketsort)	03.07.
10.7 Paralleles Sortieren	
11. Graphalgorithmen	06.07.
11.1 Durchsuchen von Graphen	
11.2 Topologisches Sortieren	
11.3 Kürzeste Wege	10.07.
11.4 Weitere Beispiele	13.07.
12. Speicherverwaltung	17.07.
12.1 Keller, Halde	
12.2 Multikeller, Garwickalgorismus	
12.3 Haldenverwaltung, Freispeicherlisten, Speicherbereinigung	20.07.
13. Netze und Prozesse	24.07.
13.1 Nebenläufigkeit, Parallelität, Petrinetze, Ereignisse	
13.2 Synchronisation, asynchrone Prozesse	27.07.
<i>-- sofern Zeit bleibt oder etwas ausgetauscht wird: --</i>	
Nachrichtenaustausch, Protokolle	
13.3 Realisierung in Ada	
13.4 Grundbegriffe zum Internet, allgemeine Dienste	
Client-Server-, "Architektur"	
Informationsdienste (Suchmaschine, Browser)	
14. Modellierung	
<i>-- Ende der Vorlesungszeit des Wintersemesters -- Vordiplom-Klausur geplant: Montag 14.8.06</i>	
Am Ende jedes Kapitels oder Abschnitts werden in der Regel historische Hinweise gegeben, manchmal auch weitere Übungsaufgaben.	

Hinweis: In den vergangenen Studienjahren entstanden jeweils bis zu 1400 Power-Point-Folien, die im pdf-Format im Netz abgelegt sind, z.B. unter

<http://www.fmi.uni-stuttgart.de/fk/lehre/ws03-04/info1/default.htm>

<http://www.fmi.uni-stuttgart.de/fk/lehre/ss05/info2/default.htm>

Diese Sammlung wird im laufenden Jahr aktualisiert. Die Folien dienen zum einen der Veranschaulichung in der Vorlesung und zum anderen der Nachbereitung und der Information für die, die nicht anwesend sein konnten. In dieses Material werden die Übungen in Zukunft stärker eingebunden, um Selbsterneffekte und Selbstkontrolle zu verbessern. Anregungen und Kritik sind willkommen.

Literatur:

Manuskripte von Prof. Claus (WS 03/04, Sommersemester ab 2002), Prof. Lagally (WS 01/02 und 04/05), Prof. Plödereder (SS 01).

Appelrath, Hans-Jürgen und Ludwig, 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), auch auf Deutsch erhältlich

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, Neuauflage 2002

Hotz, Günter, "Einführung in die Informatik", Teubner-Verlag, Stuttgart, 1992

Klaeren, H., Sperber, M., "Vom Problem zum Programm", 3. Auflage, B.G. Teubner Stuttgart und Wiesbaden, 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

Saake, G., Sattler, K.-U., "Algorithmen und Datenstrukturen", d.punkt-Verlag, Heidelberg, 2004

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

English, J., "Ada 95: The Craft of Object-Oriented Programming",

<http://www.it.bton.ac.uk/staff/je/adacraft/>

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

Riehle, R., „Ada Distilled – ...“, <http://www.adapower.com/> (- Books & Tutorials)

Weitere Literatur:

Als Einstieg: Appelrath, Boles, Claus, Wegener, "Starthilfe Informatik", Teubner-Verlag, Stuttgart-Leipzig, 2001

Für diverse Definitionen und Erläuterungen:

"Duden Informatik", dritte Auflage, Bibliografisches Institut, Mannheim, 2001;
die vierte Auflage erscheint 2006

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

Generelle Bemerkung für das Folgende:

(1) Sie müssen bis zum Ende des zweiten Semesters an der *Orientierungsprüfung* teilnehmen. Vergessen Sie die Anmeldung oder nehmen Sie nicht teil, so gelten Sie bereits als einmal durchgefallen. Bis Ende des dritten Semesters müssen Sie diese Prüfung bestanden haben, sonst können Sie in Ihrem Studienfach nicht weiter studieren. Ausnahmefälle diskutieren Sie bitte rechtzeitig mit dem Vorsitzenden des für Sie zuständigen Prüfungsausschusses!

(2) Wenn Sie merken, dass Sie die geforderten Leistungen voraussichtlich nicht erbringen können, dann setzen Sie sich *sofort* mit Ihrem Tutor, mit einem wissenschaftlichen Mitarbeiter oder mit dem Dozenten in Verbindung! Dies gilt auch bei Krankheit oder anderen Gründen, die Ihr Studium beeinträchtigen.

7. (Vordiploms-) Prüfungen und vorausgesetzte Scheine

Die Prüfungen sind für die verschiedenen Studiengänge unterschiedlich. Es wird der Stoff der Vorlesung und der Übungen geprüft. In der Regel benötigen Sie bis spätestens zum Klausurtermin den Übungsschein, siehe Punkt 8. Die Prüfung muss beim Prüfungsamt angemeldet werden (in der Regel im November bzw. im Mai).

Für die Studiengänge Informatik und Softwaretechnik ist diese Prüfung zugleich die **Orientierungsprüfung**. Diese muss im zweiten Semester angetreten werden und spätestens bis Ende des dritten Semesters bestanden sein, sonst kann das Studienfach an einer deutschen Universität nicht weiter studiert werden. Die Studierenden des Bachelorstudiengangs Wirtschaftsinformatik können alle in der Informatik erreichten Leistungspunkte auf die für ihre Orientierungsprüfung erforderlichen 40 Leistungspunkte anrechnen.

Diplomstudiengang Informatik: Die Prüfung über die Vorlesungen Informatik I und II findet als zweistündige Vordiplomsklausur "Praktische Informatik" statt (voraussichtlich am 14. August 2006, 8:30 Uhr). Hierfür sollte der Übungsschein (siehe Punkt 8) vorliegen. Falls Sie dies jedoch nicht schaffen, muss dieser Übungsschein oder der Übungsschein über die anschließende Veranstaltung "Einführung in die Informatik III" zur später zu absolvierenden Prüfung über "Informatik III" im dritten Semester vorgelegt werden. Der Schein zum Programmierkurs ist gesondert zu erwerben (siehe unten und Punkt 9).

Diplomstudiengang Softwaretechnik: Die Prüfung über die Vorlesungen Informatik I und II findet als zweistündige Vordiplomsklausur "Praktische Informatik A" statt (voraussichtlich am 14. August 2006, 8:30 Uhr). Die Teilnahme muss beim Prüfungsamt angemeldet werden. Hierfür oder spätestens zum Klausurtermin muss der Übungsschein (siehe Punkt 8 unten) vorliegen.

Bachelorstudiengang Wirtschaftsinformatik: Die Vorlesungen Informatik I und II werden in zwei getrennten Klausuren abgeprüft, und zwar als zweistündige Klausur über Informatik I (voraussichtlich am 8. März 2006, 8:30 Uhr) und als einstündige Klausur über Informatik II (voraussichtlich am 14. August 2006, 8:30 Uhr). Die Klausuren sind Prüfungsleistungen und umfassen jeweils 7 Leistungspunkte. Die Teilnahme muss beim Prüfungsamt angemeldet werden. Auch wenn der Übungsschein nicht vorausgesetzt wird, so wird dringend empfohlen, regelmäßig an den Übungen zu den Vorlesungen teilzunehmen (schließlich gehören die in den Übungen behandelten Inhalte zum Prüfungsstoff). Der Schein zum Programmierkurs ist gesondert zu erwerben (siehe unten und Punkt 9).

Diplomstudiengang Automatisierungstechnik: Die Prüfung über die Vorlesungen Informatik I und II findet als zweistündige Vordiplomsklausur "Einführung in die Informatik I und II" statt (voraussichtlich am 14. August 2006, 8:30 Uhr). Die Teilnahme muss beim Prüfungsamt angemeldet werden. Hierfür muss der Übungsschein zu "Einführung in die Informatik I (Autip)" oder "Einführung in die Informatik II" (siehe Punkt 8 unten) vorliegen.

Diplomstudiengang Computerlinguistik: Die Prüfung über die Vorlesungen Informatik I und II findet als vierstündige Vordiplomsklausur "Einführung in die Informatik I und II" statt (voraussichtlich am 14. August 2006, 8:30 Uhr). Die Teilnahme muss beim Prüfungsamt angemeldet werden. Hierfür muss der Übungsschein zu "Einführung in die Informatik I" oder "Einführung in die Informatik II" (siehe Punkt 8 unten) vorliegen.

Diplomstudiengang Mathematik mit Nebenfach Informatik: Die Prüfung über die Vorlesungen Informatik I und II findet als vierstündige Vordiplomsklausur "Einführung in die Informatik I und II" statt (voraussichtlich am 14. August 2006, 8:30 Uhr). Die Teilnahme muss beim Prüfungsamt angemeldet werden. Der Übungsschein "Einführung in die Informatik I", "Einf. i. d. Informatik II" (siehe Punkt 8 unten) oder "Einf. i. d. Informatik III" muss bis zum Abschluss der Diplom-Vorprüfung vorliegen.

Diplomstudiengang technisch orientierte Betriebswirtschaftslehre: Studierende, die „Informatik“ als Vertiefung wählen wollen, können alternativ zur zweisemestrigen Vorlesung "Grundlagen der Informatik" (Prof. Dr. O. Eggenberger) die Vorlesung "Einführung in die Informatik I" besuchen und diese prüfen lassen. Die Prüfung über die Vorlesung "Einführung in die Informatik I" findet als zweistündige Vordiplomsklausur statt (voraussichtlich am 8. März 2006, 8:30 Uhr). Die Teilnahme muss beim Prüfungsamt angemeldet werden. (Studierende nach alter Prüfungsordnung schreiben je zweistündige Klausuren zu "Einführung in die Informatik I" und "Einführung in die Informatik II".)

Informatik in Magister-, Lehramts- und anderen Studiengängen:

Diese Studierenden setzen sich bitte frühzeitig mit Herrn Dr. S. Lewandowski in Verbindung. Die Prüfung über die Vorlesungen Informatik I, II und III findet gemäß der für Ihren Studiengang gültigen Prüfungsordnung statt (voraussichtlich am 14. August 2006, 8:30 Uhr), in der Regel als zweistündige Klausur über "Einführung in die Informatik I" und "Einführung in die Informatik II" sowie als einstündige Klausur

zu "Einführung in die Informatik III". Ob zur Teilnahme ein Übungsschein vorliegen muss, entnehmen Sie bitte der für Sie gültigen Prüfungsordnung.

Nicht-Bestehen: Eine nicht bestandene Prüfung muss beim nächsten Prüfungstermin (oder innerhalb eines halben Jahres) wiederholt werden muss. Jede Prüfung können Sie nur einmal wiederholen. Falls Sie dann nochmals durchfallen, können Sie eine „mündliche Auffangprüfung im unmittelbaren zeitlichen Zusammenhang nach der Bekanntgabe des zweiten Prüfungsergebnisses“ ablegen. Wer auch diese nicht besteht, ist endgültig durch die (Diplom-) Prüfung des Faches gefallen und kann sein Studienfach an keiner deutschen Universität mehr studieren.

Programmierscheine: Unabhängig von obiger Vordiplomsklausur bzw. Prüfungsleistung über die Grundvorlesung ist in den beiden Studiengängen Informatik und Wirtschaftsinformatik der Programmierkurs zu absolvieren. Die erfolgreiche Teilnahme wird durch einen bzw. zwei Übungsscheine nachgewiesen, siehe Punkt 9 unten. Dieser Programmierschein ist im Diplomstudiengang Informatik bis zum Abschluss des Vordiploms vorzulegen. Im Studiengang Wirtschaftsinformatik sind für die Programmierübungen I und II zwei getrennte Scheine als Studienleistung zu erwerben, die jeweils 3 Leistungspunkte zählen. Insgesamt sind die Anforderungen an die Programmierübungen jedoch in beiden Studiengängen gleich. Es gelten keine Fristen, d.h., dieser Programmierschein kann beliebig oft wiederholt werden. Studierende des Diplomstudiengangs Softwaretechnik nehmen an dem Programmierkurs für Softwaretechniker teil, der von der Abteilung Software Engineering angeboten wird.

8. Übungen zur Vorlesung "Einführung in die Informatik I und II"

Die für den Übungsschein erforderlichen Leistungen bestehen aus den wöchentlich abzugebenden Übungsaufgaben (maximal 13 mal 20 Punkte im Wintersemester und 12 mal 20 Punkte im Sommersemester), insgesamt 6 Klausuren (je 3 pro Semester zu je 30 Punkten) und mindestens 3 Präsentationen pro Semester in den Übungsgruppen. Der Gesamtschein für die Diplomstudiengänge Informatik und Softwaretechnik wird am Ende der Vorlesung ausgestellt, wenn die Präsentationen erfolgreich waren und jeweils mindestens 60% der maximal erreichbaren Punkte in jedem Semester erreicht wurden (also mindestens 210 Punkte im WS und 198 Punkte im SS, Details siehe unten). Für die anderen Studiengänge werden Scheine am Ende jedes Semesters vergeben.

Die Punkte zur Erlangung des Scheins setzen sich aus zwei Leistungen zusammen: einer Gruppenleistung und einer individuellen Leistung. Alle Teilnehmer(innen) werden in *Vierer-Gruppen* (mit wenigen Ausnahmen auch Dreier- oder Fünfer-Gruppen) zusammengefasst, fünf solche Vierergruppen bilden eine Übungsgruppe. Die Übungsaufgaben werden überwiegend in den Vierergruppen bearbeitet. Die Vierergruppe löst die Übungsaufgaben und bereitet sich gemeinsam vor, die Lösungen in der Übungsgruppe vorzutragen. Die Übungsgruppenleiter wählen ein Mitglied der Vierergruppe für die Präsentation aus. Entsprechend dieser Präsentation werden die Punkte *allen* Mitgliedern der Vierergruppe angerechnet. Individuelle Punkte erhält man vor allem durch die 3 Klausuren, die Ihnen zugleich einen Hinweis geben, wie Ihr derzeitiger Wissensstand ist.

Die Übungsgruppen, die unter der Leitung eines Mitarbeiters oder Tutors stehen, finden wöchentlich zu festen Terminen statt. Jede Übungsgruppe besteht aus ca.

20 Personen. Die Teilnahme an den Übungen ist zur Erlangung des Übungsscheins verpflichtend. Es wird dringend empfohlen, an diesen Übungsgruppen aktiv teilzunehmen.

Wissenschaftliche Mitarbeiter im Wintersemester: B. Draskoczy und Dr. W. Schmid.
Studentische Tutoren im Wintersemester: A. Burkovski, D. Fischer, H. Griepentrog, V. Grigas, P. Gruber, F. Haupt, M. Krone, D. Philipp.
Vorlesung und Übungen für den Studiengang "Automatisierungstechnik in der Produktion" übernimmt im Wintersemester 05/06 Herr Dr. S. Lewandowski.

Termine: Montag 9:45 bis 11:15 Uhr, 11:30 bis 13:00 Uhr, 14:00 bis 15:30 Uhr,
Dienstag 15:45 bis 17:15 Uhr, 17:30 bis 19:00 Uhr,
Mittwoch 9:45 bis 11:15 Uhr, 11:30 bis 13:00 Uhr, 17:30 bis 19:00 Uhr,
Donnerstag 14:00 bis 15:30 Uhr, 17:30 bis 19:00 Uhr,
Freitag 9:45 bis 11:15 Uhr, 11:30 bis 13:00 Uhr.

Eintragen in die Übungsgruppen: Dies erfolgt online unter

<http://eclaus.informatik.uni-stuttgart.de/>

Der Benutzername lautet "online-Informatik1", das Passwort "claus05". Achten Sie darauf, Ihre Daten korrekt einzugeben. Insbesondere muss die E-Mail-Adresse gültig sein.

Ablauf: Die Vorlesung befasst sich in den ersten drei Wochen mit der Einführung in die Programmiersprache Ada. Die Punkte zählen ab dem zweiten Übungsblatt. Die Übungsblätter mit 2 bis 3 schriftlich abzugebenden Aufgaben, mit 2 bis 3 Votier-Aufgaben und eventuell mit einer Zusatzaufgabe liegen in der Vorlesungszeit wöchentlich ab Freitag (spätestens 17 Uhr) unter

<http://www.fmi.uni-stuttgart.de/fk/lehre/ws05-06/info1/default.htm>

im Netz, erstmals am 21.10.05. Abgabetermin ist der Samstag der folgenden Woche, 24 Uhr.

Die Übungsaufgaben zur Grundvorlesung werden so gestellt, dass ihre vollständige Bearbeitung höchstens 5 Zeitstunden für jede(n) in einer Vierergruppe beansprucht. Im Schnitt wird eine Bearbeitungsdauer von mindestens 3 Stunden je Woche von Ihnen erwartet. Fragen im Zusammenhang mit Übungsaufgaben können auch im Zusatzkurs (dienstags 8:00 bis 9:30 Uhr im Hörsaal 38.01) behandelt werden.

Zur Bearbeitung und Abgabe wird das System eClaus verwendet, das am 17.10.05 in der Vorlesung vorgestellt wird. Eine kurze Einführung ist auch im eClaus-System zu finden; dieses ist erreichbar über die Vorlesungsseite

<http://www.fmi.uni-stuttgart.de/fk/lehre/ws05-06/info1/default.htm>

Die Lösungen der Übungsaufgaben werden in den Übungsgruppen am ersten Termin besprochen, der der Abgabe folgt. Hierbei trägt je Vierergruppe ein(e) Bearbeiter(in) die Lösung der Vierergruppe vor. Stellt sich heraus, dass der/die Vortragende die Lösung nicht verstanden hat, so erhält die Vierergruppe für das gesamte Übungsblatt keinen Punkt. Die Vierergruppe muss also jedes Gruppenmitglied "fit" für den Vortrag machen.

Die korrigierten Abgaben sind in der Regel spätestens eine Woche nach Abgabetermin im eClaus-System einzusehen.

Es gibt im Wintersemester 15 und im Sommersemester 13 Übungsblätter, das erste Blatt im WS und jeweils das letzte Übungsblatt werden nicht mehr besprochen und bewertet.

Sofern Unklarheiten bestehen: Sprechstunde der Übungsgruppenleiter beachten. Erläuterungen werden ggf. ins Netz gestellt.

Fragen sowie Anregungen und Kritik können auch auf dem Schwarzen Brett

<http://swt.uni-stuttgart.de/forum/>

diskutiert werden.

Punktabzug für Abschreiben:

Liegen bei mehreren Vierergruppen offensichtlich untereinander kopierte Lösungen ("Plagiate") vor und sind die wahren Urheber nicht eindeutig zu ermitteln, werden die Punkte der betroffenen Abgabegruppen halbiert. Im Wiederholungsfall muss mit weiteren Maßnahmen gerechnet werden.

Vortrag:

Jedes Mitglied einer Vierergruppe muss mindestens drei Mal im Semester in der Übungsgruppe einen kurzen Vortrag halten, in dem die Lösung einer Übungsaufgabe vorgestellt wird. Der Vortrag wird bewertet und resultiert in einem Faktor zwischen 0 und 1.2, mit dem die auf dem Aufgabenblatt erreichten Punkte der in Gruppenarbeit bearbeiteten Aufgaben multipliziert werden. Jede Vierergruppe muss auf jedem Aufgabenblatt mindestens eine Aufgabe vorrechnen, rechnet eine Vierergruppe zwei Aufgaben vor, so werden die entsprechenden Faktoren gemittelt. Welche Gruppe welche Aufgabe vorrechnet und wer diese Aufgabe präsentiert, wird zufällig bestimmt.

Test-Klausuren während der Lehrveranstaltung:

Es finden im Wintersemester an drei Terminen (voraussichtlich am 17.11., 19.12. und 2.2. in mehreren Hörsälen) Test-Klausuren von je 35 Minuten Dauer am Ende der Vorlesung statt; im Sommersemester werden es ebenfalls drei Klausuren sein. Die Klausuren dienen zum einen dem Selbsttest, zum anderen gehen sie in die Punktzahl für den Erwerb des Übungsscheins ein.

Punktzahlen:

In jedem Übungsblatt können 20 Punkte erreicht werden. Ein Punkt entspricht rund 15 Minuten Bearbeitungszeit. In jeder Testklausur können 30 Punkte erreicht werden. Bei 13 bzw. 12 bewerteten Übungsblättern und 3 Test-Klausuren können somit 350 Punkte im Winter- und 330 Punkte im Sommersemester erreicht werden. Eine sehr gute Leistung liegt vor, wenn im Wintersemester mindestens 315 bzw. im Sommersemester mindestens 300 Punkte erreicht wurden.

Erwerb des Übungsscheins:

Ein Übungsschein wird ausgestellt, sofern

- mindestens 60% der Punkte in dem jeweiligen Semester (also mindestens 210 Punkte im WS bzw. mindestens 198 Punkte im SS) erreicht wurden,
- in den letzten vier Übungsblättern des Semesters mindestens 40 Punkte erzielt wurden,
- der/die Betreffende höchstens zwei Mal im jeweiligen Semester unentschuldig bei den Übungen gefehlt hat,
- der/die Betreffende im Semester mindestens 3 Aufgaben korrekt vorgetragen hat und durch die Beteiligung in den Übungsgruppen ersichtlich ist, dass die abgegebenen Übungsaufgaben und die Lösungen verstanden wurden.

Für die Studierenden der Informatik und Softwaretechnik wird nur ein Gesamtschein ausgestellt. Hierfür müssen jeweils in *beiden* Semestern die obigen Bedingungen erfüllt sein. (Beachten Sie die generelle Bemerkung vor Abschnitt 7!) Die Studierenden der anderen Studiengänge erhalten einen Schein pro Semester.

9. Programmierkurs, Programmierübungen

Der Programmierkurs und -übungen bestehen aus Übungen und kurzen Vorlesungsteilen nach Bedarf (donnerstags, 8:00 bis 9:30 Uhr). Die Programmierübungen sind Pflichtveranstaltung für Studierende der Informatik und der Wirtschaftsinformatik. Studierenden anderer Studiengänge wird empfohlen, diese Veranstaltung ebenfalls zu besuchen. Eine betreute Teilnahme dieser Studierenden an den Programmierübungen ist aus Kapazitätsgründen jedoch voraussichtlich nicht möglich.

Wissenschaftlicher Mitarbeiter im Wintersemester: S. Riexinger.

Studentische Tutoren für die Übungen im Wintersemester: M. Holz, M. Jakovljevic, W. Kessler, J. Kizler, P. Riegger.

Termine:

Am 17.10.05 findet im Rahmen der Grundvorlesung eine allgemeine Einführung in das eClaus-System statt. Anfangs bildet die Grundvorlesung den Einstieg in die Programmierung mit Ada. Ab der dritten Woche wird der Programmierkurs abgetrennt; die erste Vorlesung findet am 3.11. ab 8 Uhr im Hörsaal 38.01 statt. Die Übungsgruppen beginnen ebenfalls in der dritten Woche (ab dem 31.10.). Beim ersten Termin werden allerdings noch keine Übungsaufgaben besprochen, vielmehr geht es um das Sich-Kennen-Lernen, um die Erläuterung des Ablaufs und um technische Hinweise. Die Termine der Übungsgruppen:

Montag 17:30 bis 19:00 Uhr,
Mittwoch 9:45 bis 11:15 Uhr,
Donnerstag 14:00 bis 15:30 Uhr, 17:30 bis 19:00 Uhr,
Freitag 14:00 bis 15:30 Uhr, 15:45 bis 17:15 Uhr.

Die Ausgabe und Abgabe der Übungsblätter erfolgt wie bei der Grundvorlesung, jedoch mit dem Unterschied, dass die Abgabe bis zum Freitag der nächsten Woche 24 Uhr zu erfolgen hat. Das erste Übungsblatt wird am Freitag, den 28.10.05 ausgegeben. Es muss bis zum 4.11.05, 24 Uhr elektronisch abgegeben werden und wird dann in der Woche, die mit dem 7.11. beginnt, besprochen usw.

Eintragung in die Übungsgruppen: Dies erfolgt ab 17.10. online unter

<http://eclaus.informatik.uni-stuttgart.de/>

Der Benutzername lautet "online-Programmierkurs1", das Passwort "claus05". Achten Sie darauf, Ihre Daten korrekt einzugeben. Insbesondere muss die E-Mail-Adresse gültig sein.

Lernziele:

Jeder Absolvent des Programmierkurses muss in der Lage sein, eigenständig kleinere bis mittlere Probleme und Aufgaben zu analysieren, eine Lösung zu finden, diese zu formalisieren und in einen Algorithmus umzusetzen, ein zugehöriges Ada-Programm in strukturierter Form zu schreiben, zu erläutern, zu testen und eine Fehlerdiagnose und -korrektur (Debugging) durchzuführen. Besonderen Wert legen wir auf die Dokumentation, insbesondere auf eine klare Beschreibung des Lösungswegs und der gewählten Programmkonstrukte, auf lesbare, strukturierte und kommentierte Programme und auf ausgewertete Testläufe.

Aufgaben:

Es werden im Wintersemester und im Sommersemester jeweils 13 Übungsblätter (in der Regel mit je drei Aufgaben) im Programmierkurs ausgegeben, von denen

jeweils 12 bewertet werden. Für jede Aufgabe sind die Quelldateien eines lauffähigen Programms abzugeben. Prinzipiell werden nur Punkte vergeben, wenn sich das Programm fehlerfrei übersetzen lässt. Weist das Programm logische Fehler auf, so werden Punkte von der Gesamtpunktzahl abgezogen. Abhängig von der Schwere der Fehler können dies bis zu 100% der Punkte sein. Auch die Dokumentation geht stark in die Bewertung ein.

In der Quelldatei des Hauptprogramms ist am Anfang eine Herleitung des Programms als Kommentar einzufügen. Fehlt diese Herleitung oder ist sie nicht schlüssig, können Punkte abgezogen werden. Dies gilt ebenfalls für schlechten Programmierstil und unstrukturierte Notationen.

Falls im Aufgabentext Dateibezeichner für die Quelldateien oder spezielle Formate für Ein- und Ausgaben des Programms vorgegeben werden, sollten Sie sich exakt an diese Vorgaben halten. Fehlerhafte Ein- und Ausgabeformate werden als logische Fehler interpretiert.

Die grundsätzliche Lösung der Aufgaben können Sie gerne in Gruppen diskutieren. **Allerdings muss das lauffähige Programm von jedem Studierenden selbst ohne fremde Hilfe erstellt und implementiert werden.** Insbesondere erwarten wir individuelle Bezeichner, Strukturierungen und Kommentare. Wir werden die Ähnlichkeit von Programmen überprüfen und bei identischen oder sehr ähnlichen Abgaben keine Punkte für die beteiligten Studierenden vergeben und im Wiederholungsfall weitere Maßnahmen ergreifen.

Die Abgabe der Quelldateien für die einzelnen Aufgaben erfolgt im eClaus-System

<http://eclaus.informatik.uni-stuttgart.de>

Ablauf einer Übungsstunde:

In den Übungsstunden werden Lösungen von den Studierenden vorgestellt und erläutert. Hier werden auch offene Fragen zu den Aufgaben geklärt.

Erwerb des Übungsscheins:

Für den Erwerb des Übungsscheins gelten folgende Bedingungen:

- Mindestens 50% der Punkte müssen erreicht werden, d.h. wenigstens 120 Punkte aus den bewerteten 12 Übungsblättern.
- In den letzten 4 Übungsblättern müssen mindestens 40 Punkte erreicht werden.
- Der Besuch der Übungen ist Pflicht. Unentschuldigtes Fehlen ist höchstens zwei Mal im Semester zulässig.
- Wenigstens zwei Mal muss an der Tafel ein Lösungsvorschlag vorgetragen und erläutert werden.

Die Studierenden der Wirtschaftsinformatik erhalten einen Schein pro Semester. Dieser zählt als Studienleistung mit 3 Leistungspunkten. Für die Studierenden der Informatik wird ein Gesamtschein ausgestellt, sofern jeweils in beiden Semestern die obigen Bedingungen erfüllt wurden.

Die Scheine enthalten die erreichte Punktzahl und die zugehörige Note.

Die Note lautet:

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

der maximal erreichbaren Punkte.

Inhaltliche Planung für das Wintersemester: bis Weihnachten klassischer PASCAL-Teil, danach Graphen und Pakete.

Überblick über den Inhalt von Programmierkurs/-übungen im Wintersemester:

- Organisatorisches, Rechner, (Ada-) Software, Beispiele
- Programmierrichtlinien, Dokumentation, Fehler
- Datenstrukturen, strenge Typisierung/Anpassung
- Programmieren mit Funktionen
- Blöcke, Semantik von Prozeduraufrufen, Parameterübergabe
- Pointer, Listen, Halde
- Bäume, Durchlaufalgorithmen
- Pakete, Spezifikations- und Implementierungsteil
- Beispiele, Import/Export, Programmierung von Graphen

Grobplanung für das Sommersemester: Ausnahmebehandlung, Vertiefungen bei dynamischen Datenstrukturen, Klassenkonzept in Ada, Nebenläufigkeit (tasks), größere Beispiele (insbesondere zu Graphen) und Programmierprinzipien.

10. Zu guter Letzt der gute Rat:

Die Erfahrungen in den letzten Jahren haben uns gezeigt, dass die Studierenden weniger mit den Inhalten als vielmehr mit ihrer falschen oder nicht vorhandenen Arbeitsweise Probleme haben. Planen Sie Ihre Arbeit! Reservieren Sie in der Vorlesungszeit rund 42 Zeitstunden (einschl. der Veranstaltungsbesuche, aber ohne Transportzeiten) für die eigentliche Arbeit in Ihrem Studium. "Arbeit" bedeutet im ersten Studienjahr vor allem: Wiederholen und Verstehen des Vorlesungsstoffs, Lösen der Übungsaufgaben zum einen in einer Vierergruppe (Übungen zur Grundvorlesung) und zum anderen individuell (Programmierübungen). Entwickeln Sie Techniken, wie Sie den Stoff am besten lernen (nicht einfach nur Text lesen; da bleibt zu wenig im Kopf hängen; nutzen Sie die Diskussion in Ihrer Gruppe). Überwachen Sie Ihre Arbeitszeit: Schreiben Sie genau auf, wieviel Zeit Sie für was aufwenden und korrigieren Sie ggf. Ihre Handlungsweisen. Teilen Sie die Arbeitszeit ein; sitzen Sie nicht zu oft am Rechner oder an den reinen Programmierarbeiten! Diskutieren Sie den Stoff und die Aufgaben mit anderen, um selbst festzustellen, wo Sie stehen, was Sie gut oder nicht verstanden haben, wo Sie noch lernen und üben müssen usw. Werden Sie sich selbst gegenüber kritisch. Ab sofort müssen Sie sich selbst zum (sinnvollen und zielgerichteten) Lernen befähigen.

Ihr Merkzettel: Zusammenfassung des Organisatorischen

Veranstalter: Die Veranstaltungen "Einführung der Informatik I und II" und "Programmierübungen I und II" werden im Studienjahr 2005/2006 durchgeführt von der Abteilung Formale Konzepte (FK) im Institut für Formale Methoden der Informatik (FMI). Dieses Institut gehört zur Fakultät 5 "Informatik, Elektrotechnik und Informationstechnik" der Universität Stuttgart. Postanschrift: Abteilung Formale Konzepte, Universität Stuttgart, Universitätsstraße 38, 70569 Stuttgart.

Veranstaltungstermine:

Montag: Vorlesung 15:45 - 17:15 Uhr, Hörsaal 38.01, Start 17.10.05
(auch: Austeilen der Übungsaufgaben in Papierform)

Dienstag: Zusatzkurs 8:00 - 9:30 Uhr, Hörsaal 38.01, Start 25.10.05

Donnerstag: Programmierkurs (bei Bedarf) 8:00-9:30 Uhr, Hörsaal 38.01

Donnerstag: Vorlesung 15:45 - 17:15 Uhr, Hörsaal 38.01 (ab 20.10.05)

Freitag: Bereitstellung der nächsten Übungsaufgaben im Netz bis 17 Uhr

Freitag: Abgabeschluss der Übungen zum Programmierkurs 24:00 Uhr

Samstag: Abgabeschluss der Übungen zur Grundvorlesung 24:00 Uhr

Ihre Übungsgruppe zur Grundvorlesung:
(selbst eintragen)

Ihre Übungsgruppe zum Programmierkurs:
(selbst eintragen)

Einzeltermine (geplant, die Termine können sich noch ändern):

17.10.05	In der Vorlesung: Einführung in das System eClaus
18.10.05	8:00 - 8:45 Uhr: Hinweise zum Studium der Informatik
03.11.05	8:00 Uhr, Beginn Programmierkurs
28.11.-2.12.	Anmeldung zu den Klausuren im Februar/März 2006
17.11.05	am Ende der Vorlesung: Test-Klausur
12.12.05	am Ende der Vorlesung: Lehrevaluation
19.12.05	am Ende der Vorlesung: Test-Klausur
02.02.06	am Ende der Vorlesung: Test-Klausur
18.02.06	Ende der Vorlesungszeit des Wintersemesters
08.03.06	8:30 Uhr, Vordiplomsklausur (für höhere Semester)
08.03.06	8:30 Uhr, Klausur über Informatik I für Wirtschaftsinformatik
24.04.06	15:45 Uhr: Beginn der Grundvorlesung im Sommersemester
Mai 06	Anmeldung zu den Klausuren für August bis Oktober 06
29.07.06	Ende der Vorlesungszeit des Sommersemesters
14.08.06	Abschlussklausur für alle Studiengänge (insbes. Vordiplom)

Verantwortliche Personen: (Name, Raum, Telefon: 0711 / 7816 - ..., E-Mail)
Prof. Dr. Volker Claus, Raum 1.111, -300, claus@informatik.uni-stuttgart.de
Dipl.-Inf. Botond Draskoczy, Raum 1.162, -225, draskoczy@informatik.uni-stuttgart.de
Dr. Stefan Lewandowski, Raum 1.108, -336, lewandowski@informatik.uni-stuttgart.de
Dipl.-Inf. Sascha Riexinger, Raum 1.101, -337, riexinger@informatik.uni-stuttgart.de
Dr. Wolfgang Schmid, Raum 1.108, -401, wolfgang.schmid@informatik.uni-stuttgart.de

Sekretariat der Abteilung Formale Konzepte:
Heike Photien, Raum 1.117, Tel.: 0711 / 7816 - 328, photien@informatik.uni-stuttgart.de

Sprechstunden des wissenschaftlichen Personals:

Claus: in der Regel Montag 13-14 Uhr, Raum 1.111 oder nach Vereinbarung
Draskoczy: nach Vereinbarung, Raum 1.162
Lewandowski: Donnerstag 14:45-15:45 Uhr, Raum 1.108
Riexinger: Mittwoch 13:30-14:30 Uhr, Raum 1.101
Schmid: vormittags nach Vereinbarung, Raum 1.108

2. Intro. DEUTSCH Bedienungsanleitung bitte lesen und aufbewahren! Nicht wegwerfen! Bei Schäden durch Bedienungsfehler erlischt die Garantie! Technische Änderungen vorbehalten! ENGLISH Please read and retain these directions for use. Do not throw them away! The warranty does not cover damage caused by incorrect use of the equipment! Leicht zu lesen. Here you will find a selection of literature easy to read. (B2 Level) to choose from the stock of our library. To facilitate your search, we have marked the books with an adhesive "Leicht zu lesen". Aehnlich, Kathrin Alle sterben, auch die LÄffelstÄrre : Roman 2009. - 256 S. -The story of a special lifelong friendship, wonderfully tender and sad at the same time Translation for 'wichtiges Dokument' in the free German-English dictionary and many other English translations. GermanDies ist ein sehr wichtiges Dokument, und ich bedaure wirklich, daŸ sich hier nicht mehr Mitglieder an der Diskussion beteiligen. more_vert. open_in_new Link to source. warning Request revision. This is a very important document and indeed I am sorry that there are not more Members here to discuss it. GermanHerr PrÄsident, liebe Kolleginnen und Kollegen, der Bericht unseres Kollegen Aldo ist in meinen Augen ein sehr wichtiges Dokument. more_vert. open_in_new Link to source.