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

Einführung in die Informatik I und II

Studienjahr 1.10.02 bis 30.9.03

1. Zur Informatik

Informatik befasst sich mit dem Rohstoff „Information“: der Darstellung, Verarbeitung, Speicherung, Übertragung usw, insbesondere aus technischer Sicht. Da die menschlichen Bereiche durch Informationen gesteuert werden, hat sie nachhaltigen Einfluss auf unsere Gesellschaft, auf das Arbeitsleben und auf das Privatleben jedes einzelnen. Stichwörter sind die Automatisierung geistiger Abläufe, die Beherrschbarkeit von Prozessen und die Methoden zur Bearbeitung unüberschaubarer Vorgänge. Wesentlich ist hierbei, dass die Objekte und Produkte der Informatik nicht „anfassbar“ sind.

Für das Verständnis der Informatik sind die *Abstraktion* und die *Formalisierung* besonders wichtig. Abstraktion bezeichnet das Herausarbeiten gemeinsamer Prinzipien aus den unterschiedlichsten (Anwendungs-) Gebieten, z.B. die Darstellung von Strukturen in Programmiersprachen, von syntaktischen Gebilden bei natürlichen Sprachen oder von Rechenbäumen mit Hilfe von (meist kontextfreien) Grammatiken. Bereits die Repräsentation von Objekten der realen Welt durch Buchstaben und Sprachen sowie durch andere Informationen bildet eine Abstraktion, wodurch an Informatiker(innen) wesentlich höhere Abstraktionsanforderungen gestellt werden als beispielsweise an „klassische“ Ingenieure. Abstraktes Denken ist eine wichtige Voraussetzung für eine adäquate *Modellierung*, durch die Dinge der realen Welt in Programmiersprachen abgebildet werden, wodurch in der Regel eine automatisierte Bearbeitung erst möglich wird. Formalisierung ist die Fähigkeit, Geschehnisse der realen Welt oder (unscharfe) Informationen zu präzisieren und in Kalkülen zu notieren. Hierdurch werden sie der Verarbeitung durch einen Computer zugänglich und es können Informationen zwischen Menschen in eindeutiger Weise ausgetauscht werden. Zugleich lassen sich zu formalisierten Objekten oft Aussagen wie Laufzeit, Speicherplatzbedarf, Korrektheit usw. gewinnen, die anderenfalls höchstens grob geschätzt werden könnten.

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 (vor allem: Zeitkomplexität)
- Datenstrukturen und die Effizienz zugehöriger Algorithmen,
- Herangehensweisen zur Problemlösung, insbesondere die
- Entwicklung von Algorithmen und Datenstrukturen einschl. Verifikation und Analyse,
- Realisierung und Implementierung der Konzepte,
- Abstraktion und Theorie, auf denen diese Begriffe basieren,
- genaue Kenntnis wichtiger Standardalgorithmen,
- genaue Kenntnis typischer Beispiele und ihre Programmierung,
- genaue Kenntnis mindestens einer Programmiersprache.

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

3. Struktur der Veranstaltung

Die Veranstaltung besteht aus (1 „Stunde“ sind hier 45 Zeit-Minuten):
Vorlesung: 4 Vorlesungsstunden pro Woche,
Übungen: 2 Übungsstunden pro Woche,
Programmierkurs: 0,5 Stunden Vorlesung je Wochen (ab 28.10.),
Programmierübungen: 1,5 Übungsstunden je Woche (ab 28.10.).
Programmierkurs und Programmierübungen sind Pflicht für Studierende der Informatik und der Wirtschaftsinformatik. Allen anderen wird empfohlen, den Vorlesungsteil des Programmierkurses mitzuhören.

4. Anforderungen

Wir erwarten von Ihnen regelmäßige Mitarbeit, und zwar beträgt der studentische Aufwand für die gesamte Lehrveranstaltung (4 Stunden Vorlesung, 2 Stunden Übungen und 2 Stunden Programmierkurs/-übungen pro Woche) wöchentlich 17 bis 20 Zeitstunden für Studierende der Informatik und Wirtschaftsinformatik bzw. 13 bis 15 Zeitstunden für alle anderen Hörer(innen). Im Einzelnen:
- 3 Zeitstunden für den Besuch der Vorlesung
- 1,5 Zeitstunden für die Teilnahme an der Übungsgruppe
- 1,5 Zeitstunden für die Teilnahme an Programmierkurs und -übungen
- 4 bis 5 Stunden Nachbereitung des Vorlesungsstoffs (einschl. Lesen in Büchern)
- 3 bis 4 Stunden Bearbeitungsdauer der Übungsaufgaben.
- 3 bis 5 Stunden Bearbeitungsdauer der Programmieraufgaben.

Wird diese Zeit regelmäßig (!) aufgewendet, so reichen weitere 40 bis 60 Stunden zur Prüfungsvorbereitung aus.
Für Personen, die weitere Beispiele kennen lernen möchten oder die den Stoff nochmals erläutert haben wollen, wird zusätzlich ein „Stützkurs“ angeboten.

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.

5. Voraussetzungen

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. Als Programmiersprache wird Ada 95 verwendet; Programmierkurs/-übungen dienen der „handwerklichen“ Vertiefung in dieser Sprache. In der Vorlesung wird für die Algorithmen zusätzlich eine vereinfachte Ada-ähnliche Notation verwendet, die auch Basis für Semantik, Verifikation und Komplexität ist. Viele Algorithmen werden direkt in Ada 95 ausformuliert.

6. Vorlesungsinhalte im Einzelnen

Die Vorlesung folgt inhaltlich den Vorlesungen der vergangenen Jahre, die von Prof. Lagally und Prof. Plödereder gehalten wurden und als Skript im Netz oder bei der Fachschaft zur Verfügung stehen. Wie im jährlichen Rhythmus üblich werden Teile umgestellt und gewisse Aspekte anders betont. Einige Algorithmen werden elektronisch veranschaulicht; sie werden danach im Netz bereitgestellt. Für die Verwaltung und Abgabe der Übungen wird auf das Intranet zurückgegriffen.

Umfang der Vorlesung: 29 Wochen à 4 Vorlesungsstunden (1 VLS = 45 Minuten). Davon fallen rund 6 wegen Feiertagen usw. aus. Zur Verfügung stehen somit 110 Vorlesungsstunden. Wir unterteilen diese in drei Abschnitte und Vorbemerkungen.

Geplante Gliederung der Grundvorlesung:

0. Vorbemerkungen

0.1 Zum Studium
- 0.0.1 Planung im Studium, Selbst-Überwachung
- 0.0.2 Ihre Arbeitsleistung, Arbeitsmoral
- 0.0.3 Ziele des Studiums, Ziele der Vorlesung
- 0.0.4 Voraussetzung für diese Vorlesung
- 0.0.5 Ablauf und Ihre Mitwirkung
- 0.0.6 Zusammensetzung der Vorlesung
- 0.0.7 Bücher, Skript, Folienkopien, Mitschrift
- 0.0.8 Umstellung auf „elektronisches Textbuch“
- 0.0.9 Unerwünschtes
- 0.0.10 Fragen?

0.2 Hinweise zu den Unterlagen, zur Durchführung, zu Übungen und Programmierkurs

0.3 Formalismen (*diese Teile werden irgendwo nach Bedarf in die Vorlesung integriert!*)
- Mathematische Grundsymbole
- Mengen, spezielle Mengen, Darstellungen (auch Zahldarstellungen)
- Relationen, Funktionen, Codierungen
- Ausdrücke (Terme)
- Beweise, Induktion
- Gleichungen, Identitäten

1. Grundlagen der Programmierung (Wintersemester, ca. 15. Oktober bis 10. Januar)

1.1. Algorithmen und Sprachen
- Darstellung von Algorithmen, grundlegende Sprachelemente
- Grundlegende Datenbereiche
- Realisierte Abbildung
- Einfache künstliche Sprachen
- Regeln, Grammatiken, Sprachen (auch: BNF, Syntaxdiagramm)
- Sprachen zur Beschreibung von Sprachen, speziell von Algorithmen

1.2. Aussagen über Algorithmen
- Charakteristika von Algorithmen
- Grenzen der Algorithmen, Unentscheidbarkeit
- Korrektheit von Algorithmen (axiomatischer Ansatz, schwächste Vorbedingung)
- Komplexität von Algorithmen, O-Notation
- Darstellung durch Funktionen und Gleichungen

1.3. Daten und ihre Strukturierung
- Elementare Daten
- Konstruktoren ($\times$, $\cup$, Wortmenge, Potenzmenge, Funktionenmenge)

Relationen, Graphen, Referenzen (gerichtete Kanten)
Speziell: Listen, Bäume
Beschreibung in einer Sprache

1.4. Grundbegriffe der Programmierung (wird mit 1.5 verzahnt)
Aufbau von Programmen (Bezeichner, Deklarationen, sequentiell, verteilt, parallel, Stile)
Elementare Datenbereiche, elementare Anweisungen
Daten-Konstruktoren, Anweisungs-Konstruktoren
Umgebungen (lokal, global, Gültigkeit, Lebensdauer)
Deklaration von Algorithmen (Prozeduren, Funktionen)
speziell: Parameterübergabe und Kopierregel
Abstrakter und konkreter Datentyp
Relationen auf Datentypen (z.B. Vererbung)
Objekte und Klassen
Programmiersprache und ihre Beschreibungssprache
(Hinweise auf Interpreter, Übersetzer)

1.5. Die Sprache Ada 95 (wird weitgehend mit 1.4 verzahnt)
Aufbau von Programmen
Elementare Datenbereiche, elementare Anweisungen (einschl. Ein-/Ausgabe)
Daten-Konstruktoren, Anweisungs-Konstruktoren
Blöcke (kellerartig verwaltete Datenbereiche)
Deklaration von Algorithmen (Prozeduren, Funktionen)
Module
Generizität
Weitere Sprachelemente

1.6. Semantik von Programmen
(Speicher-) Zustand, Durchlauf durch Programme, Berechnung
Berechnete Funktion (determiniert, deterministisch usw.)
Transitionssystem
Ein-Schritt-Semantik und Ergebnissemantik (Regelsystem)
Anwenden auf Korrektheit, Verifikation, „Beweise“
Spezifikation

1.7. Komplexität von Algorithmen und Programmen (sofern Zeit bleibt)
Rechenmodell, Maschinen
Präzisierung der Komplexität
Rechenstruktur, Maschinen über Datenstrukturen

2. Interaktionen (Ende des Wintersemesters, ca. 11. Januar bis 15. Februar)

2.1. Objektorientierung
Datentyp, Vererbung, Zuständigkeit, Zustand, Klasse, Instanz/Ausprägung usw.

2.2. Prozesse
Nebenläufigkeit, Parallelität, Ereignisse
Synchronisation, asynchrone Prozesse
Nachrichtenaustausch, Protokolle
Realisierung in Ada

2.3. Vernetzte Systeme
Verteilte Algorithmen, verteilte Daten
Client-Server-„Architektur“
Internet (Adressierung, Protokoll, Sprache HTML, Dienste, www)
Informationsdienste (Suchmaschine, Browser)

2.4. Modellierung (sofern Zeit bleibt)
Modellierung von Daten (ER)
Modellierung von Abläufen (Ablaufdiagramme, Zustandsdiagramme)
Modellierung komplexerer Zusammenhänge (angelehnt an UML)

3. Grundlegende Verfahren (Sommersemester, ca. 20. April bis 20. Juli)

3.1. Verwaltung von Datenstrukturen
Elementare Typen
Felder (arrays)
Pointer, Listen
Stapel, Warteschlangen (Unterprogramme)
Dynamische Daten (Halde)

3.2. Suchen
Suchen in flachen Strukturen
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 (und Sondierungen)
Analyse der Hashverfahren
Rehashing

3.4. Sortieren
Überblick, allgemeine Aussagen
Aussuchen, Auswählen (Min., Heapsort)
Einfügen (Listen, Bäume, Radix, Shell)
Austauschen (Bubblesort, Quicksort)
Mischen (Mergesort und diverse Varianten)
Streuen und Sammeln (Bucketsort)
Paralleles Sortieren

3.5. Graphalgorithmen
Überblick, Definitionen
Durchsuchen von Graphen
Topologisches Sortieren
Kürzeste Wege
Minimale Spannbäume
Zusammenhangskomponenten
Flüsse in Netzen

3.6. Geometrische Algorithmen (sofern Zeit bleibt)
Datenstrukturen
Schnittproblem für Rechtecke
Konvexe Hülle
Minimale Entfernungen
Eine Sprache für die Elementargeometrie

Am Ende jedes Kapitels oder Abschnitts werden in der Regel historische Hinweise gegeben.

Hinweis: Bis zum Juli 2004 soll es zu der Vorlesung eine Beamer-Präsentation und somit ein „elektronisches Textbuch“ geben. In dieses Textbuch werden die Übungen stärker eingebunden, um Selbstlerneffekte und Selbstkontrolle zu verbessern. Daher wird die Nutzung elektronischer Medien in den kommenden Jahren stetig zunehmen. Die Umstellung erfolgt in den Jahren 2002 bis 2004.

Literatur:

Manuskripte von Prof. Lagally, WS 01/02 und Prof. Plödereder, SS 01
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 2002
Hotz, Günter, "Einführung in die Informatik", Teuber-Verlag, Stuttgart, 1992
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

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

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

7. (Vordiploms-) Prüfungen

Die Prüfung findet als Klausur statt. Die Teilnahme muss beim Prüfungsamt angemeldet werden. Voraussetzung ist in der Regel ein Schein.
Über Informatik I findet eine Klausur Ende Februar statt, über Informatik II sowie über beide Teile zusammen (Vordiplome in Informatik, Softwaretechnik, Wirtschaftsinformatik, Nebenfach Informatik in anderen Studiengängen) voraussichtlich Anfang August 2003.
Derzeit gibt es folgende Varianten betreffend die Vorlesungen „Einführung in die Informatik I, II, III“ für die verschiedenen Studiengänge:

1a. Orientierungsprüfung für die Diplomstudiengänge Informatik und Softwaretechnik, zugleich Prüfung in Praktischer Informatik (1. Teil über Informatik I, II im Vordiplom): 2 Stunden Klausur.
1b. Praktische Informatik (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. Praktische Informatik A (Informatik I und II), Vordiplom Softwaretechnik: 2 Stunden, kann für die Orientierungsprüfung verwendet werden.
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), kann für die Orientierungsprüfung verwendet werden.
10. Wahlfachprüfung eventuell (Physik, Technische Kybernetik, Magisterstudiengänge, ...): ?

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 Auffangprüfung im unmittelbaren zeitlichen Zusammenhang nach der Bekanntgabe des zweiten Prüfungsergebnisses“.

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

8. 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 15 bis 20 Personen. Es wird dringend empfohlen, an diesen Übungsgruppen aktiv teilzunehmen. Es werden voraussichtlich 18 Gruppen eingerichtet.

Wissenschaftliche Mitarbeiter (in Klammern die Zahl von ihnen betreuten Übungsgruppen) im Wintersemester: J. Holub (2), S. Lewandowski (1), W. Schmid (1).
Tutoren im Wintersemester: F. Braun (2), B. Faßbender (2), S. Göller (2), S. Grottel (2), P. Jehlicka (2), H. Kahl (2).

Termine: Montag 10:15 bis 11:45 Uhr, 13:15 bis 14:45 Uhr, 16:45 bis18:15 Uhr,
Dienstag 13:15 bis 14:45 Uhr, 15:00 bis 16:30 Uhr, 16:45 bis 18:15 Uhr,
Mittwoch 10:15 bis 11:45 Uhr, 13:15 bis 14:45 Uhr,
Donnerstag 13:15 bis 14:45 Uhr

Es gibt im Wintersemester 15 und 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/inf1/groups`
Der Benutzername lautet "inf1", das Passwort "claus02". 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/ifi/fk/lehre/ws02-03/info_I_0203.html`
im Netz, erstmals am 18.10.02.
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 Donnerstag (erstmals am 24.10.02), 13:00 Uhr elektronisch, d.h. Bearbeitungszeit nach Ausgabe: 6 Tage.
Zur Bearbeitung und Abgabe wird das System eClaus verwendet, das am 17.10.02 in der Vorlesung vorgestellt wird. 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 zwei 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 Wintersemester an drei Terminen (voraussichtlich am 21.11, 19.12. und 30.01) Testklausuren im Umfang von je 30 Minuten Dauer im Rahmen der Vorlesung statt; im Sommersemester werden es ebenfalls drei Klausuren sein. 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 370 Punkte im Winter- und 310 Punkte im Sommersemester erreicht werden. Eine sehr gute Leistung liegt vor, wenn im Wintersemester mindestens 333 bzw. im Sommersemester mindestens 279 Punkte erreicht wurden.
Sofern im WS mindestens 185 Punkte bzw. im SS mindestens 155 Punkte erreicht werden, der/die Betreffende im Semester mindestens 2 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.

9. Programmierkurs, Programmierübungen

Wissenschaftlicher Mitarbeiter im Wintersemester: S. Schwoon (2).
Tutoren für die Übungen im WS: T. Kern (2), J. Kizler (2), F. Schlachter (2).

Diese Veranstaltung ist nur für die Studierenden der Informatik und der Wirtschaftsinformatik verpflichtend. Er besteht überwiegend 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 12 Übungsblätter ausgegeben, so dass 240 Punkte erreicht werden können. Die Regelungen von Abschnitt 8 gelten analog für die Ausstellung dieses Scheins (nur 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: Der Programmierkurs wird in der dritten Woche mit einer Vorlesungsstunde am 28.10.02, 16:45 - 17:30 Uhr, im Hörsaal 20.01 starten.
Der Übungsbetrieb beginnt in der gleichen Woche zunächst mit technischen Hinweisen. In dieser Woche wird auch das erste Übungsblatt ausgegeben..
Die Übungen für Informatiker finden am Dienstag, 8:30 Uhr, und Donnerstag, 8:30 Uhr, die für Wirtschaftsinformatik am Donnerstag, 16:45 Uhr, statt.

Eintragen in die Übungsgruppen: Es erfolgt online unter
`https://inf2.informatik.uni-stuttgart.de/uebungsgruppen-bin/infpk/groups`
Der Benutzername lautet "infpk", das Passwort "claus02". Achten Sie darauf, Ihre Daten korrekt einzugeben. Insbesondere muss die Email-Adresse gültig sein.

Inhaltliche Planung des Wintersemesters: bis Weihnachten klassischer PASCAL-Teil, danach dynamische Datenstrukturen und Moduln.

Überblick über den Inhalt von Programmierkurs/-übungen:

- Organisatorisches, Rechner, erste kleine Beispiele.
- Programmaufbau, Elementare Datenbereiche, Ausdrücke, Variable, Wertzuweisungen, Ein-/Ausgabeanweisungen, Deklaration
- Anweisungen (;, if, case, for, while, …), Felder
- Deklaration von Prozeduren und Funktionen, Parameter
- Programmierrichtlinien, Dokumentation, Fehler
- Datenstrukturen, strenge Typisierung/Anpassung
- Programmieren mit Funktionen
- Blöcke, Semantik von Prozeduraufrufen
- Pointer, Listen, Halde
- Bäume, Durchlaufalgorithmen
- Pakete, Spezifikation
- Beispiele, Vertiefungen, Ausnahmebehandlungen
- Beispiele, Vertiefungen, Import/Export

Grobplanung des Sommersemesters: Klassenkonzept in Ada, Nebenläufigkeit (tasks), größere Beispiele und Programmierprinzipien, etwas Software-Engineering.