JOHANN WOLFGANG GOETHE – UNIVERSITÄT
INSTITUT FÜR INFORMATIK

PROF. DR. DETLEF KRÖMKER

Grundlagen der Programmierung 1
Einführung in die Programmierung

PRG 1 und EPR

WS 16/17

Institut für Informatik

Graphische Datenverarbeitung – Prof. Dr. Detlef Krömker

Robert-Meyer-Straße 10 • 60054 Frankfurt am Main.

Telefon +49 69/798 24600• Fax +49 69/798 24603

kroemker@cs.uni-frankfurt.de – http://www.gdv.cs.uni-frankfurt.de

VORWORT

Dieses Skript begleitet die Vorlesungen und Übungen der Veranstaltung „Grundlagen der Programmierung 1“ und „Einführung in die Programmierung“. Es erhebt nicht den Anspruch, ein Lehrbuch zu sein, dessen Studium die Veranstaltungen zu ersetzen vermag. Es soll nicht statt, sondern neben der Vorlesung verwendet werden.

Das Modul PRG 1 mit den Veranstaltungen:

PRG 1: 2V + 2Ü (2 Vorlesungs- + 2 Übungsstunden) pro Semesterwoche,
und EPR: 1V + 2Ü (1 Vorlesungs- + 2 Übungsstunden) pro Semesterwoche.

ergeben zusammen 11 Kreditpunkte/CP und repräsentieren damit 37% des Stoffs dieses Semesters.

Die Veranstaltungen PRG 1 und EPR sind also formal geteilt: In PRG1 werden eher die konzeptionellen Eigenarten des Programmierens besprochen (häufig das WARUM?); in EPR das praktische Programmieren mit Python (das WIE?). Trotzdem bilden diese beiden Veranstaltungen eine Einheit und sind in dem Modul „Programmierung 1“ zusammengefasst. Die Kreditpunkte sind ein Maß für den sogenannten *Workload* (1CP entspricht 30 Stunden Arbeitsbelastung für einen durchschnittlichen Studierenden).

Es gelten folgende Nebenbedingungen:

1. Aus den bearbeiteten (Haus-)Übungen können bis zu 40% der zum Bestehen notwendigen Punkte als (Bonus-)Punkte als Klausurpunkte übernommen werden. Die „Bonuspunkte“ stammen aus beiden Übungen PRG1 und EPR. Angerechnet wird die gewichtete „Gesamtpunktzahl“ aus den Übungen PRG1 und EPR und den Quizzes. Insgesamt werden mindestens 120% an Hausübungspunkten gestellt, man muss also nicht bei jeder Aufgabe 100% der möglichen Punkte erreichen, um den maximalen Bonus zu erhalten.

2. Wir wollen Ihre aktive Teilnahme an den Übungen. In der PRG1-Übung muss jeder Teilnehmer mindestens einmal „vorgerechnet“ haben, sonst gilt die Übung als nicht erfolgreich und die erreichten Übungspunkte werden für die Klausur auf NULL gesetzt.

3. Die Studierenden müssen **mindestens 50 ~~% der~~ Übungspunkte in EPR erzielt haben**, um das Modul PRG1 erfolgreich abzuschließen. Man kann die Modulprüfung ablegen, aber das Modul wird erst dann als erfolgreich im Prüfungsamt gewertet, den die EPR-Bedingung (Studienleistung) erfüllt ist.

Sie müssen als „Handwerkszeug“ das Programmieren erlernen – es geht kein Weg daran vorbei. Sie brauchen diese Programmierkenntnisse nicht nur in der Prüfung sondern in vielen anderen Bereichen: z.B. auch zum Nachvollziehen der Theorie: „Modellierung“ oder „Algorithmen und Datenstrukturen“ im Sommersemester, usw. – kurz: Ohne Programmieren geht es nicht! – Wir werden Ihnen schon in diesem Semester diese Verbindungen aufzeigen.

In diesem Skript werden die grundlegenden Konzepte der Programmierung vorgestellt. Diese einführende Veranstaltung ist für Studierende folgender Studiengänge konzipiert:

- Bachelor Informatik
- Bachelor Bioinformatik
- L3 und L2/L5 Informatik (Lehramt)
- Master Wirtschaftsinformatik (wenn Bachelorstudium nicht Informatik)
- Kognitive Linguistik – Schwerpunkt Computerlinguistik
- Bachelor Physik der Informationstechnik
- Nebenfach Informatik in Diplom-, Magister- und Bachelor- und
- Masterstudiengängen anderer Fachbereiche

Als Einstiegssprache wird Python gewählt, es folgen Haskell und SQL in PRG 2 und JAVA im Programmierpraktikum. Der Umfang der behandelten Gebiete richtet sich nach den Empfehlungen des Fakultätentages Informatik, nämlich in PRG 1 und EPR:

Teil 1: Was ist Informatik? – eine basale **Einführung in die Programmierung** [Krömker]: Grundlegende Elemente und Konzepte imperativer und objektorientierter Sprachen: Datenstrukturen, Kontrollstrukturen, Datentypen; strukturiertes Programmieren (5 CP)

Teil 2: Objektorientierte Programmierung und Design – Warum, Wieso?, Paradigmen der OO-Orientierung, OO-Entwurf (2CP)

Teil 3: Elemente der Softwaretechnik (des Software Engineerings) [Ramesh]: Anforderungen, Spezifikation, Korrektheit, Testen, Dokumentation; Entwicklungszyklen, Modularisierung ; vom Problem zum Algorithmus: Algorithmenentwurf. (2CP)

Teil 4: Rechnernetze und Verteilte Systeme [Kisel]: Dienste und Protokolle, Kommunikationssysteme, Internet, Netzarchitekturen und Netzsicherheit; Prozesse, Nebenläufigkeit, Synchrnisation und Kommunikation, **Nutzung von Betriebssystemen** aus Sicht einer Programmiersprache: Aufgaben und Struktur, Dateien und Dateisysteme, Sicherheit und Schutzmechanismen.

(2CP)

PRG 1+ EPR - Summe 11 CPs

Ausblick oder Rückblick: In PRG 2 (im Sommersemester)

Teil 1: Übersicht Sprachparadigmen [Schmidt-Schauß]: Funktionale Programmierung am Beispiel Haskell, Rekursion vs. Iteration, Typisierung, Compilerbau, Operationale Semantik für funktionale Programmiersprachen,

(4 CPs)

Teil 2: Einführung in Datenbanksysteme [Zicari]: Architekturen, Konzeptionelle und logische Modelle, Entity-Relationship-Modell, Relationenmodell, Normalformen, Datenbankdesign, Abfragesprachen (SQL)

(4 CPs)

PRG 2 - Summe 8 CPs

Ergänzt werden diese Veranstaltungen durch das Programmierpraktikum im 3. Oder 4. Fachsemester des Informatikstudiums (8 CPs), das insbesondere in folgenden Bereichen eine fachliche Ergänzung darstellt:

Programmierung (im „Größeren“, im Projekt)	(2 CP)
Grundlagen der Betriebssysteme	(2 CP)
Grundlagen der Softwaretechnik (Praxis)	(2 CP)
Rechnernetze oder Verteilte Systeme	(2 CP)
	Programmierpraktikum - Summe 8 CP

Insgesamt entsprechen diese Veranstaltungen dem Pflichtkatalog gemäß der Empfehlungen des Fakultätentages Informatik für den Bachelor Informatik.

Das Curriculum ist streng modular aufgebaut – und ist auch ohne Informatik-Vorkenntnisse zu erarbeiten. Vorkenntnisse sind natürlich hilfreich, sie verkürzen ggf. den Lernaufwand für einzelne Themen – **aber Vorsicht**: Verpassen Sie nicht wichtige einzelne Themen oder gar den Anschluss, wenn Sie anfangs das Gefühl haben, alles sei ganz einfach.

Jede Lerneinheit wird in 1½ Stunden in einer Vorlesung vorgestellt – da wir schon Ende der vorletzten Semesterwoche die Klausur schreiben, fällt der Freitagstermin nur gelegentlich aus. Sie sollten jede Vorlesung besuchen (mindestens die Aufzeichnungen ansehen) und ggf. nacharbeiten. Die Quizzes geben Ihnen Hinweise, wieviel Sie und was Sie verstanden haben. Die Zeit in der Vorlesung ist für die Menge des Stoffs sehr knapp – im Wesentlichen können wir hier nur die Rolle eines „Guides“ übernehmen und den Stoff **systematisch darstellen** - oft nur die essentiellen Gedanken entwickeln und nicht alle Details vortragen. Insofern spielt die Vorlesung eine wichtige Rolle für Sie. Sie zeigt deutlich, was **essenziell** und was „nur“ wichtig ist. Das Skript bietet zum Teil detailliertere oder vertiefende Informationen, die auch Prüfungsstoff sind und im Selbststudium erarbeitet werden sollen. Manche Details (insbesondere) der Programmiersprache werden als aufgezeichnete Kurzvorlesungen in der Lernplattform bereitgestellt (häufig ca. 15 Min lang) – die im „Inverted Classroom“- Modus behandelt werden, d.h, in der Präsenz werden hierzu vertiefende Fragen besprochen. Haben Sie die Videos nicht angeschaut, haben Sie das zugehörige Quiz nicht durchgeführt, haben Sie Garnichts von der Präsenzveranstaltung. SORRY.

Zu jeder Präsenzvorlesung und jeder Kurzvorlesung gibt es ein Quiz (einen eKurs): Dies sind Sammlungen von Fragen zu dem Stoff der Vorlesung. Es wird Ihnen sehr stark angeraten, diese Kurse zu absolvieren und **nicht nur** als Klausurvorbereitung zu nutzen.

In diesem Skript verzichten wir auf die Anwendung der strengen wissenschaftlichen Zitierregeln – bei diesen Grundlagen wäre dies sehr umfänglich und der Text durch viele Fußnoten ggf. viel schwerer verständlich, würden wir die strenge wissenschaftliche Form waren. Dieses Skript ist also in diesem Sinn kein wissenschaftlicher Text!

Gegenüber den Vorjahren wurde dieses Skript teilweise deutlich überarbeitet, und korrigiert sowie fehlende Teile ergänzt. Was geblieben ist, ist sein Umfang: Jede Lehreinheit umfasst etwa 25 Seiten. Ich hoffe, dass eine zum Teil ausführlichere Darstellung Sie dazu verleitet, sich mit den Themen intensiv auseinanderzusetzen. Bitte beachten Sie: Die zum Teil umfangreichen Anhänge und die vielen Internet-Links sind wirklich nur als Ergänzung für vertiefende Studien gedacht oder geben Details

wieder, die im Haupttext das wesentliche „verdecken“ könnten. Diese müssen Sie gerade beim ersten Durcharbeiten nicht unbedingt lesen!

Detlef Krömker, Oktober 2016

Inhalt – Vorlesung 0

Vorlesung

0

Begrüßung und Einführung

Ziel dieser Vorlesung ist es, Ihnen einen Einstieg in die Informatik als Wissenschaft zu geben – im Allgemeinen und speziell an der Goethe-Universität. Wie lernt man Informatik, welche Veranstaltungen gibt es, welche Materialen stehen für Sie zur Verfügung? Auch das Thema Scheine und Prüfungen muss geklärt werden. Und abschließend einige „Specials", die speziell für diese Veranstaltung entwickelt wurden .

1 WIE LERNT MAN INFORMATIK

1.1 UNIVERSITÄRES LERNEN UND LEHREN

Die Universität ist die älteste Form einer wissenschaftlichen Einrichtung für Forschung und Lehre, mit folgenden Grundsätzen:

- *universitas magistrorium et scholarium* (Gemeinschaft der Lehrer und Schüler)
- *universitas litteratum* (Gesamtheit der Wissenschaften)
- Einheit von Forschung und Lehre
- Autonomie und Selbstverwaltung (Freiheit von Forschung und Lehre, Art. 5 Grundgesetz)
- dem Recht, akademische Grade zu vergeben

Die ersten Universitäten wurden im 11. Jahrhundert in Italien gegründet (streng genommen noch Vorläufer der Universitäten, insbesondere Rechtsschulen in Ravenna, Bologna, Padua). Danach im 12. Jhd. die Universität zu Paris (Sorbonne), im deutschen Sprachraum im 14. Jahrhundert (kurz nacheinander Prag, Wien, Heidelberg, Erfurt).

Was bedeutet studieren?

- *studere* (sich bemühen)
- *studium* (die Mühe, das Bemühen)

Ein wesentliches Merkmal des Studiums ist **Selbstständigkeit.** – Auch wenn wir uns sehr bemühen, Ihnen den Einstieg ins Studium so einfach wie möglich zu machen. Lernen müssen Sie.

Trotzt allen Reformen, insbesondere in den letzten 50 Jahren, ist das universitäre Leben immer noch von vielen Traditionen geprägt, u.a.

Die Einteilung des Jahres in Semester:

Wintersemester (1. Oktober bis 31. März) mit einer Vorlesungszeit (dieses Mal 15. Oktober 2012 bis 15. Februar 2013 = 15 Wochen)

Sommersemester (1. April bis 30. September) mit einer Vorlesungszeit (SS 12: 10. April 2012 bis 13. Juli 2011 = 13 Wochen)

wird hessenweit festgelegt, zurzeit bis 2018 (siehe http://www.uni-frankfurt.de/org/ltg/admin/lsf/vorl-zeiten.html).

- Zeitangaben mit c.t. (*cum tempore* = mit Zeit) = 15 Minuten nach der vollen Stunde,
  Zeitangaben mit s.t. (*sine tempore* = ohne Zeit) = „pünktlich" (zur vollen Stunde).

Zum Selbstverständnis der Goethe-Universität: Die **Johann Wolfgang Goethe-Universität** Frankfurt am Main ist eine lebendige, urbane und weltoffene Universität. Die Gründung als Stiftungsuniversität im Jahre 1914 verdankt die Universität der für die Stadt Frankfurt charakteristischen Verbindung von dynamischer Wissenschaft, dem Erbe der Aufklärung und selbstbewusstem bürgerlichen Engagement auf der Grundlage von internationalem Handel und Industrie**: Bürgeruniversität** (Goethe trägt kein **von**).

Als eine der größten Universitäten in Deutschland mit mehr als 45.000 Studierenden und mehr als 600 ProfessorInnen bietet die Universität ein breites Fächerspektrum in Forschung und Lehre. Die Goethe-Universität und auch das Institut für Informatik orientieren sich an den „Grundsätze[n] zu Lehre und Studium an der Goethe-Universität". [1] Dort heißt es:

„UNIVERSITÄRE LEHRE UND FORSCHENDES LERNEN

*Die Goethe-Universität bekennt sich zur Idee der **Einheit von Forschung und Lehre** sowie der individuellen Entwicklung durch Wissenschaft. Sie stellt sich der Herausforderung einer zeitgemäßen Reformulierung dieser Universitätsprinzipien, schafft die hierfür erforderlichen Rahmenbedingungen und gestaltet ihre Studienangebote so, dass sie von Studierenden auch bei unterschiedlichen Voraussetzungen erfolgreich absolviert werden können.*

*Als Charakteristikum universitärer Lehre betont die Goethe-Universität **Wissenschaftlichkeit,** die in der didaktisch-methodischen Konzeptualisierung des forschenden Lernens ihren Ausdruck findet. Studierende erfahren Wissen hierbei als etwas Offenes, im Werden Begriffenes – eben als Forschung. Sie werden so zu Fragenden*

[1] Siehe: http://www.uni-frankfurt.de/51044043/Grundsaetze-Lehre-Studium.pdf

und zu Forschenden, die von Anfang an befähigt werden sollen, Wissen und seine Entstehungsbedingungen kritisch zu hinterfragen.

***Forschendes Lernen heißt Identifizieren und Kontextualisieren von Problemlagen**, beinhaltet stets Skepsis und die Fähigkeit zu distanzierter Betrachtung, bildet Selbständigkeit und methodisch angeleitete Urteilsfähigkeit aus. Es setzt bei den Studierenden Begeisterung für das Fach und Eigenverantwortung voraus. Universitäre Lehre an der Goethe-Universität vermittelt Fach- und Methodenwissen, das die Grundlage für den Erwerb wissenschaftlicher Kernkompetenzen ist. Zugleich werden die Studierenden in die Lage versetzt, konkret am wissenschaftlichen Gegenstand überfachliche Kompetenzen einzuüben. Dazu gehört nicht zuletzt die Befähigung, sich selbständig Wissen und Informationen anzueignen. Die Absolvierung eines Studiums eröffnet somit sowohl den Weg in die Forschung als auch zu akademisch-qualifizierter Berufstätigkeit, was bei einigen Studiengängen die konkrete Berufsvorbereitung bedeuten kann. …".*

1.2 ZIEL DES STUDIUMS BACHELOR OF SCIENCE INFORMATIK

In diesem Abschnitt wird auf das Bachelor Studium Informatik fokussiert. Informationen und Regelungen für andere Studiengänge findet man unter:

http://www-extern.informatik.uni-frankfurt.de/Informatik/studium-1/nebenfach-informatik-1

Aus der Ordnung des Bachelor-Studiengangs Informatik (http://www.informatik.uni-frankfurt.de/images/pdf/informatik/bachelor2/bachelorordnung_neu.pdf):

„§ 2 Ziel des Bachelorstudiengangs

(1) Der Bachelorstudiengang ist ein selbstständiger Studiengang, der zugleich der erste Abschnitt eines konsekutiven Studiums der Informatik mit nachfolgendem Masterstudiengang ist. Der Bachelorstudiengang ist grundlagen- und methodenorientiert und legt somit die Grundlagen des Faches Informatik in der Breite. Er stellt sicher, dass die Voraussetzungen für spätere Verbreiterungen, Vertiefungen und Spezialisierungen im Fach gegeben sind. Er bereitet insbesondere auf das Masterstudium vor. Der Bachelorstudiengang soll dazu befähigen, die vermittelten Fähigkeiten und Kenntnisse anzuwenden und sich im Zuge eines lebenslangen Lernens schnell neue, vertiefende Kenntnisse anzueignen. Er ermöglicht einen Einstieg in den Arbeitsmarkt für entsprechende Aufgaben oder den Wechsel des Studienorts.

(2) Dieser Studiengang befähigt die Absolventen und Absolventinnen durch seine Grundlagenorientierung zu erfolgreicher Tätigkeit über das gesamte Berufsleben hinweg, da er sich nicht auf die Vermittlung aktueller Inhalte beschränkt, sondern theoretisch untermauerte grundlegende Konzepte und Methoden vermittelt, die über aktuelle Trends hinweg Bestand haben.

(3) Die Ausbildung vermittelt den Studierenden die grundlegenden Prinzipien, Konzepte und Methoden der Informatik. Die Absolventen und Absolventinnen sollen nach Abschluss ihrer Ausbildung insbesondere in der Lage sein, Aufgaben in verschiedenen Anwendungsfeldern unter gegebenen technischen, ökonomischen und sozialen Randbedingungen mit den Mitteln der Informatik zu bearbeiten, entsprechende Systeme zu entwickeln und Projekte zu leiten. Sie sollen die erlernten Konzepte und Methoden auf zukünftige Entwicklungen übertragen können. Exemplarisch sollte Einblick in ein Anwendungsfach genommen werden.

(4) Problemlösungskompetenz: Die Absolventen und Absolventinnen sollen im Stande sein, komplexe Aufgaben systematisch und mit Informatikmethoden zu spezifizieren, brauchbare und zuverlässige Lösungen zu konstruieren und diese zu validieren. Sie sollen bei auftretenden Problemen Maßnahmen ergreifen können, die zu deren Lösung notwendig

sind. Die Absolventen und Absolventinnen sollen darin geübt worden sein, unüberschaubar scheinende Fragestellungen konstruktiv in Angriff zu nehmen. Sie müssen gelernt haben, hierfür Systeme und Techniken der Informatik zielorientiert einzusetzen.

(5) Schlüsselqualifikationen und Interdisziplinarität: Neben der technischen Kompetenz sollen die Absolventen und Absolventinnen Konzepte, Vorgehensweisen und Ergebnisse kommunizieren und im Team arbeiten können. Sie sollen im Stande sein, sich in die Sprache und Begriffswelt der Anwender und Anwenderinnen einzuarbeiten, um über Fachgebietsgrenzen hinweg zusammenzuarbeiten. Sie sollen grundlegende Erfahrung im Projektmanagement und Führungsqualifikation und Managementkompetenz erworben haben. Diese Qualifikationen können unter anderem im Ergänzungsmodul erworben werden.

(6) Auswirkungen der Informatik: Die Absolventen und Absolventinnen sollen die Auswirkungen der Informatik auf die Gesellschaft in ihren sozialen, wirtschaftlichen, arbeitsorganisatorischen, psychologischen und rechtlichen Aspekten einschätzen können. Ihnen sollen die ethischen Leitlinien für die Berufsausübung bewusst sein.

(7) Die Fähigkeiten von Absolventen und Absolventinnen dieses Studiengangs lassen sich durch die folgenden Prädikate charakterisieren:

> *1. Die Absolventen und Absolventinnen beherrschen die mathematischen und informatischen Methoden, Probleme in ihrer Grundstruktur zu analysieren.*
>
> *2. Die Absolventen und Absolventinnen beherrschen die informatischen Methoden, abstrakte Modelle aufzustellen.*
>
> *3. Die Absolventen und Absolventinnen haben gelernt, Probleme zu formulieren und die sich ergebenden Aufgaben in arbeitsteilig organisierten Teams zu übernehmen, selbstständig zu bearbeiten, die Ergebnisse anderer aufzunehmen und die eigenen Ergebnisse zu kommunizieren.*
>
> *4. Die Absolventen und Absolventinnen haben die methodische Kompetenz erworben, um programmiertechnische Probleme insbesondere auch im Kontext komplexer Systeme unter ausgewogener Berücksichtigung technischer, ökonomischer und gesellschaftlicher Randbedingungen erfolgreich bearbeiten zu können.*
>
> *5. Die Absolventen und Absolventinnen sind sich der vielfältigen Sicherheitsprobleme bewusst, die mit dem Einsatz von Informatiksystemen, insbesondere im Netz, verbunden sind; sie wissen, welche Techniken und Verfahren für die Sicherung von Systemen zum Einsatz kommen.*
>
> *6. Die Absolventen und Absolventinnen haben exemplarisch ausgewählte Anwendungsfelder kennen gelernt und sind in der Lage, bei der Umsetzung informatischer Grundlagen auf Anwendungsprobleme qualifiziert mitzuarbeiten.*
>
> *7. Die Absolventen und Absolventinnen haben exemplarisch außerfachliche Qualifikationen erworben und sind damit für die nichttechnischen Anforderungen und erforderlichen Sozialisierungsfähigkeit im betrieblichen Umfeld sensibilisiert.*
>
> *8. Die Absolventen und Absolventinnen sind durch die Grundlagenorientierung der Ausbildung gut auf lebenslanges Lernen und auf einen Einsatz in unterschiedlichen Berufsfeldern vorbereitet.*

(8) Der Bachelorstudiengang Informatik an der Johann Wolfgang Goethe-Universität wird gebildet aus den Fachgebieten:

Informatik der Systeme
Grundlagen der Informatik
Angewandte Informatik

Zum Bachelorstudiengang Informatik an der Johann Wolfgang Goethe-Universität gehören weiterhin Veranstaltungen unter anderem des Instituts für Mathematik, in denen die wichtigsten Grundkenntnisse, Beweisverfahren und Arbeitstechniken der Mathematik vermittelt werden, soweit sie für die Informatik von Belang sind. Das Studium umfasst auch Veranstaltungen zur Reflexion über gesellschaftliche Auswirkungen der Informatik. Darüber hinaus muss ein Anwendungsfach gewählt werden, das eine Anwendung von Informatik-Methoden und -Techniken ermöglicht und benötigt.

Das Lehrangebot des Anwendungsfaches besteht aus Lehrveranstaltungen im Gesamtumfang von 24 CP (vgl. §20) und beinhaltet Lehrveranstaltungen im Umfang von in der Regel 4 CP, in denen die Anwendungen von Methoden und Techniken der Informatik Gegenstand sind. Es bestehen Anwendungsfachvereinbarungen mit Betriebswirtschaftslehre, Biologie, Chemie, Geographie, Geophysik, Kognitive Linguistik, Mathematik, Medizin, Meteorologie, Philosophie, Physik, Psychologie und Volkswirtschaftslehre.

Der Zugang zu einem Anwendungsfach kann zahlenmäßig beschränkt sein. Die Auswahl der Studierenden richtet sich in diesem Fall nach den Bestimmungen des für das Fach zuständigen Fachbereichs.

Andere Fächer können als Anwendungsfächer durch den Fachbereichsrat des Fachbereichs Informatik und Mathematik nach Maßgabe von §20 Abs. 7 zugelassen werden.“

1.3 ROLLE UND STRUKTUR DER VORLESUNGEN

Zentrales Element und Anliegen der Vorlesungen PRG1 und EPR ist, Ihnen das Programmieren nahe zu bringen.

Die Zeit in der Vorlesung ist knapp – im Wesentlichen kann ich hier nur die Rolle eines „Guides“ übernehmen und oft nur die essentiellen Gedanken entwickeln, nicht alle Details vortragen. Insofern spielt die Vorlesung eine wichtige Rolle für Sie, sie zeigt deutlich, was essenziell und was „nur“ wichtig ist.

Die Vorlesungen dienen der systematischen Vorstellung des Stoffs. Das Skript ist entsprechend der Vorlesungen gegliedert. Details entnehmen Sie bitte dem Dokument

Vorlesungsübersicht (im Moodle-Kurs)

Sie sollten **jede Vorlesung besuchen und nacharbeiten**, indem Sie mindestens **die Quizzes** zu diesem Teil durcharbeiten und ggf. dieses Skript (nach-)lesen oder besser sogar, die Vorlesung mit dem Skript vorbereitet haben. Die Quizzes sind Sammlungen von Fragen, die den Vorlesungstoff umfassen. Durch die Bearbeitung dieser Fragen erhalten Sie etwas mehr Sicherheit, ob Sie den Stoff der Vorlesung auch verstanden haben.

Die Vorlesungen werden aufgezeichnet und werden Ihnen zur Verfügung gestellt.

Eine Vorlesung ist keine „Show“, auch wenn ich bemüht sein werde, diese kurzweilig zu gestalten! Eigentlich ist die Vorlesung eine überholte Lehrform aus dem Mittelalters, als **mitschreiben** notwendig war, weil es sonst keine Materialien gab. Dies kann auch heute noch sinnvoll sein – sinnvoll sind auf jeden Fall persönliche Zusammenfassungen.

Prinzip: **Ich bin „nur“ Lotse – nicht Schullehrer!**

Unser gemeinsames Ziel: Informatische Bildung!

- Ich habe den Stoff ausgewählt, die Reihenfolge, die Methodik, …
- Sie versuchen bitte, mir zu folgen … geben mir Feedback!
- Wir werden Sie auch auf die Prüfungen gezielt vorbereiten.

Aber, **LERNEN** müssen Sie!

1.4 ROLLE UND STRUKTUR DER ÜBUNGEN

Die Übungen finden in Kleingruppen statt, ca. 25 Studierende und werden von einer TutorIn geleitet. Tutoren sind erfahrene und gute Studierende, die dieses Modul schon erfolgreich absolviert haben:

Die Übungen sind **Präsenzübungen:** Es werden jeweils kleinere Präsenzübungsaufgaben behandelt, die zur Lösung der Hausübung hinführen sollen. – für Kleingruppen gibt es jeweils 5-10 Minuten Überlegungszeit und dann sollten die Lösungsideen vorgestellt werden. Dies ist die notwendige Beteiligung für das „Vorrechnen“. Dies wird in der Moodle Plattform notiert und angezeigt. Die zurückliegenden Hausübungen werden nur im Bedarfsfall auf Anforderung behandelt!

Details zur Gruppeneinteilung siehe Folien und Moodle-Kurs. Insgesamt sind es in PRG und EPR zusammen vier Übungsstunden pro Woche. Für PRG gibt es jeweils ein Übungsblatt pro Woche (insgesamt mindestens 12) für EPR 3 Einzelübungen und 5 zwei-wöchentliche Übungen im Pair-Programming (als Zweiergruppe). Die Mitglieder einer Zweiergruppe müssen in einer Übungsgruppe sein.

Die Einzel-Aufgaben sind in den Hausübungen individuell zu bearbeiten. Es gilt: Werden **Dubletten (Kopien, Plagiate)** gefunden, so wird dies als Betrugsversuch gewertet!

Es gibt es beim ersten Feststellen für eine(n) Studierende(n) **keine Punkte für dieses Übungsblatt, für niemanden! – KEINE Diskussionen. Wir vergleichen die Abgaben elektronisch und prüfen im Verdachtsfall persönlich nach – dann steht die Entscheidung im ersten Fall!**

Achtung: Beim zweiten Feststellen gibt wird eine persönliche Rücksprache beim Prüfer (Prof. Dr. Detlef Krömker) angeboten und ggf. der Betrugsfall festgestellt: Daraus folgt dann der Verlust der Anrechnung jeglicher Übungspunkte in der Klausur!

Die Ausgabe der Übungsblätter erfolgt bis spätestens Mittwochs; Abgabe bis Freitags 9.30 Uhr (für PRG1) respektive Samstag 16 Uhr (EPR) , also 9 bis 11 Tage Bearbeitungszeit. Falls nötig erfolgt die Besprechung erfolgt dann in der Übungsstunde der folgenden Woche!

Alle Abgaben der Übungsblätter haben elektronisch zu erfolgen (getippt ;-)) ganz zur Not eingesanned (Scanner steht bei der RBI)

Über die Lernplattform Moodle können Sie ggf. Tipps zur Lösung erfahren (aber bitte **keine** Musterlösung). Aber bitte erst auf dem FAQ (Frequently Asked Questions) Server nachschauen. Sie haben **keinen** Anspruch auf eine Antwort!

1.5 DIE BEREITGESTELLTEN MATERIALIEN

Alle Materialien und Hinweise, also insbesondere

- dieses Skript (und weitere Readings)
- eLecture: eine Aufzeichnung der Vorlesung als Videostream oder zum Download.
- Quizzes + Programmierhandzettel und Progrmmierhandbuch
- die Vorlesungsfolien als .pdf.
- Übungsaufgaben (und ggf.exemplarische Lösungen)
- diverse Specials

finden Sie auf der Lernplattform Moodle unter:

https://moodle.studiumdigitale.uni-frankfurt.de/moodle2/course/view.php?id=325

Hier sollten Sie unbedingt regelmäßig nachschauen, was es Neues gibt. Wir stellen das Skript und die anderen Materialien Zug um Zug zur Verfügung, weil wir einen „Druck-Kollaps“ im RBI vermeiden wollen.

1.6 ERGÄNZENDE LITERATUR

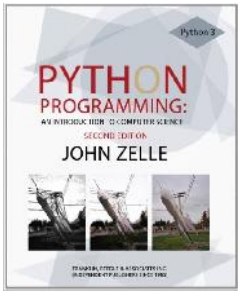
Natürlich gibt es viele Bücher und Angebote im Netz, die den Stoff dieser Veranstaltung abdecken. Gäbe es ein ideales Buch, so hätten wir dieses Skript **nicht** geschrieben. Viele Bücher beziehen sich auf andere Programmiersprachen oder verfolgen gänzlich andere Einführungsstrategien, also: Das „ideale" Material ist wohl dieses Skript. Aber es gibt Ergänzungen, die interessant sein könnten. Die nachfolgende Auswahl ist subjektiv und garantiert unvollständig.

Wir gliedern dies einmal nach

- Bücher – Allgemeine Einführungen in die Informatik
- Bücher zu Python
- Python Kurse und sonstige freie Angebote im Internet

Englisch ist die **lingua franca** (Verkehrssprache) der Informatik. Auch der Python-Interpreter wird nur Englisch mit Ihnen sprechen. Dies ist der Grund, warum wir auch englische Quellen angeben, die zumindest für Python fast immer die aktuellere Version darstellen … und viele Dokumente werden überhaupt nicht übersetzt. Spätestens zum Abschluss Ihres Bachelorstudiums sollten Sie das Fachenglisch gut beherrschen … und das gelingt nur, wenn man es übt.

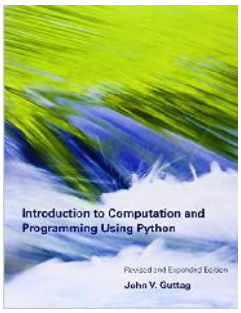
Bücher: Allgemeine Einführungen in die Informatik (dies sind also Einführungen, die auch wesentliche Teile von PRG1 mit abdecken):



John Zelle: Python Programming: An Introduction to Computer Science

Taschenbuch: 528 Seiten, Verlag: Transatlantic Publishers; Auflage: 2nd Revised edition. August 2010); ISBN-10: 1590282418; ISBN-13: 978-1590282410

ca. 34 €



John V. Guttag: Introduction to Computation and Programming Using Python; Taschenbuch: 296 Seiten; Verlag: The Mit Press; Auflage: Exp Rev (2. August 2013); ISBN-10: 0262525003; ISBN-13: 978-0262525008

ca. 25 €

Hierzu gibt es einen frei zugänglichen MOOC (Massive Open Online Course) vom MIT (zusammen mit Eric Grimson und Ana Bell), siehe bei edX:

https://www.edx.org/course/introduction-computer-science-mitx-6-00-1x-5
(Registrierung nötig)

Hans Peter Gumm, Manfred Sommer: Einführung in die Informatik, Oldenbourg, Zehnte Auflage, 2012; 912 Seiten, ISBN-10: 3486706411;
ISBN-13: 3486706411

Auszüge auch im Internet:
http://www.informatikbuch.de/ (leider nur unvollständig)

ca. 45 €

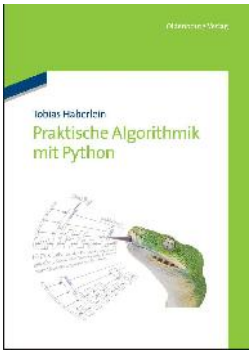
Kommentar: Gutes Buch, benutzt aber JAVA als Einstiegssprache.

Tobias Häberlein: Eine praktische Einführung in die Informatik mit Bash und Python, September 2011; Oldenbourg Wissenschaftsverlag; 213 Seiten;

SBN-10: 3486704230, ISBN-13: 978-3486704235.

Ca. 30 €

Kommentar: sehr knapp gefasst. 1. Auflage hat viele Typos, aber für diese Veranstaltung durchaus geeignet.



Tobias Häberlein: Praktische Algorithmik mit Python; Juli 2012; Oldenbourg Wissenschaftsverlag; 331 Seiten; ISBN-10: 3486713906;
ISBN-13: 978-3486713909;

Ca. 40 €

Kommentar: Etwas theoretisch angehaucht, aber dadurch das für alle Algorithmen auch Python-Implementierungen angegeben sind, für EPR (grenzwertig) empfehlenswert. … Es gelten übrigens dieselben Anmerkungen wie für diesen Autor oben angegeben.

Zu Python: Achtung - Wir benutzen die aktuelle Release 3.5.2

Zu Python findet man nun wirklich fast alles im World Wide Web. **Die zentrale Anlaufstelle ist**

python™

www.python.org dann weiter

die **Python 3.5.2 documentation** unter **https://docs.python.org/3/** . Hierdrauf werden Sie früher oder später sowieso zurückgreifen müssen.

Das Original Python Tutorial ist recht gut, siehe das aktuelle Python-Tutorium in Englisch.

- http://docs.python.org/py3k/tutorial/

Das aktuelle Python-Tutorium in Deutsch gibt es nur für die Version 3.3 NICHT 3.4 (die Unterschiede sind für **Sie nicht relevant** … es ist also gut brauchbar – die Übersetzung noch o.k.)

- https://py-tutorial-de.readthedocs.org/de/python-3.3/ (HTML)
  https://bitbucket.org/cofi/py-tutorial-de/downloads (.pdf und .epub)

Python ist sehr populär geworden … es gibt buchstäblich hunderte von Angeboten im Web. Aber bitte passen Sie auf und nehmen Sie keine Materialien der Version 2.X (vor 2012), das könnte Sie in manchen Dingen verwirren. Hier eine sehr subjektive Auswahl.

Freie eKurse im Web

Im Web gibt es unzählige Angebote zum Python Lernen – freie genauso wie kommerzielle – leider sehr viel Mist, aber auch ein paar gute Angebote. Auf sehr wenige freie Angebote möchte ich hinweisen.

Ein guter Kurs (WBT) **für Informatik-EinsteigerInnen** vom Computer Science Circle.

http://cscircles.cemc.uwaterloo.ca/0-de/

Der Python Interpreter muss nicht unbedingt installiert werden; interessante Visualisierung des Programmablaufs mit einem Werkzeug von Philip (MIT & U Rochester) … also Programmieren im Browser.

Eine deutsche Übersetzung des amerikanischen Originals http://cscircles.cemc.uwaterloo.ca/der U Waterloo.

Martin v. Löwis: Spielend Programmieren lernen! - Ein vierwöchiger **MOOC** (Massive Open Online Course,) des HPI … auch hier, wenn man will: Programmieren im Browser.

https://open.hpi.de/courses (dann Kurs auswählen und registrieren) Neuer Kurs: 9. November 2015 - 7. Dezember 2015

Sehr viel weniger als bei uns!

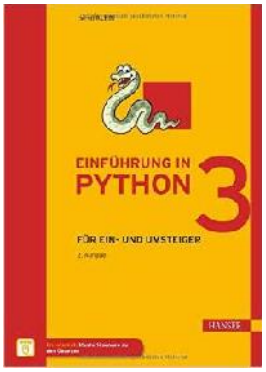
Auch für Umsteiger und Fortgeschrittene ist der Kurs von

Bernd Klein: **Python 3 Tutorial:**
http://www.python-kurs.eu/python3_kurs.php

Hier finden Sie auch zusätzliches Material zu fortgeschrittenen Themen und z.B. TKinter. Der Autor dieses freien Web-Angebotes hat auch ein Buch zu Python 3 geschrieben:

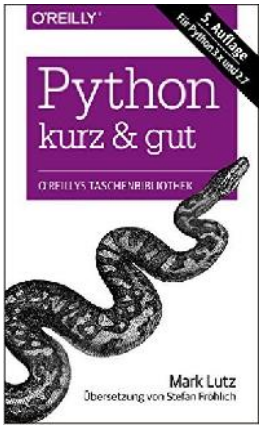
Kommentar: Es entspricht in etwa unserem Niveau – ist also gut geeignet zur Begleitung.

Bücher



Klein, Bernd: Einführung in Python 3 - Für Ein- und Umsteiger; 2. überarbeitete und erweiterte Auflage. 10/2014; 512 Seiten; Verlag: Carl Hanser Verlag GmbH & Co. KG; ISBN-10: 3446441336; ISBN-13: 978-3446441330;

Preis: ca. 25 €.



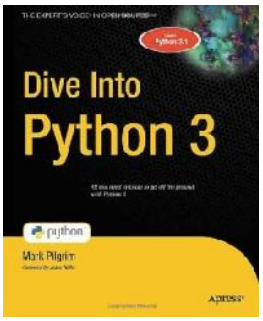
Mark Lutz: Python - kurz & gut (übersetzt von Stefan Fröhlich), Taschenbuch: 280 Seiten Verlag: O'Reilly Verlag GmbH & Co. KG; Auflage: 5 (Mai 2014) ISBN-10: 395561770X, ISBN-13: 978-3955617707

Als Nachschlagewerk sehr gut! Erklärt nur „WIE".

Preis: ca. 15 €

Die Originalversion in Englisch: Mark Lutz: Python Pocket Reference, 5th Edition, O'Reilly Media January 2014. 266 pages. Updated for both Python 3.4 and 2.7.

Interessant einmal anzuschauen: About Python, http://www.rmi.net/~lutz/about-python.html



Mark Pilgrim: **Dive Into Python 3**

In Englisch frei im Netz verfügbar http://www.diveintopython3.net/

Wie der Autor schreibt: *Also available on dead trees!*
2nd Edition *ISBN-13: 978-1430224150 ISBN-10: 1430224150*

Preis: ca. 27 €

In Deutsch:

Pilgrim, Mark: Python 3 – Intensivkurs; übersetzt aus dem Amerikanischen von Florian Wollenschein.; **1st Edition**., 2010, XVIII, 500 S., Softcover;
Springer ISBN: 978-3-642-04376-5

z.Zt. auch frei verfügbar: http://www.springer.com/us/book/9783642043765
Preis: ca. 60 €

und viele, viele weitere Quellen. (Sorry an die Autoren, die ihre Werke hier nicht berücksichtigt finden)

Weitere Literatur wird ggf. noch bekannt gegeben. Es lohnt sich auch immer, einmal unter

Tech books for free download http://www.techbooksforfree.com/perlpython.shtml .

nachzuschauen, ob es frei verfügbare Bücher gibt!

2 SCHEINE UND PRÜFUNGEN

2.1 LEISTUNGSNACHWEIS IN EPR

Es müssen mindestens **50 Übungspunkte** (von 120) aus den EPR Hausübungen erreicht worden sein. Dies können Sie beliebig oft versuchen.

In den Präsenzübungen muss mindestens einmal vorgerechnet worden sein. Diesen Fakt erheben die Tutoren.

2.2 MODULPRÜFUNG „PROGRAMMIEREN 1 - 11 CP“

Die Modulprüfung findet als Klausur statt, am **Freitag, den 3.2.2017, 10-14Uhr**; die Nachklausur am Mittwoch, 12.4.2016 (Osterwoche).

Hier gibt es mehrere Boni:

1. Durch die Klausuraufgaben können 100 Punkte erworben werden. Zum Bestehen Note = 4.0 benötigt man mindestens 50 Punkte. Aus korrigierten Hausübungen (PRG1 + EPR, siehe oben) sind in diesem Fall 40% von 50 Punkten, also **maximal 20 Punkte anrechenbar**.

2. Es gibt **Freiversuche** (§ 34 der Ordnung) für alle Basismodule (also PRG 1), solange die Prüfung im vorgesehenen Fachsemester abgelegt wird, also Studienanfänger im SoSe sind jetzt im zweiten Fachsemester oder Studienanfänger im WiSe sind jetzt im ersten Fachsemester. **Freiversuch** heißt: Mit „nicht ausreichend" bewertete Prüfungsleistungen gelten als **nicht unternommen.**

3. **Bestandene Modulabschlussprüfungen** können einmal zur **Notenverbesserung** wiederholt werden, wobei die bessere Leistung angerechnet wird. Von dieser Regelung darf höchstens zweimal im Bachelorstudiengang Gebrauch gemacht werde

4. Nicht bestandene Prüfungsleistungen können bis zu zweimal, bzw. dreimal bei geltend gemachtem Freiversuch, wiederholt werden.

5. Wiederholungsprüfungen zu B-PRG1 sind innerhalb von 15 Monaten abzulegen. Überschreitung der Wiederholungsfristen wird als Nichtbestehen gewertet.

Noch ein Hinweis für Informatikstudierende: Wer im ersten Semester erfolglos studiert (kein Schein, keine Modulprüfung bestanden), ist verpflichtet eine Studienberatung in Anspruch zu nehmen!

3 SPECIALS UND WEITERE HINWEISE

3.1 PEER-MENTORING + LERNGRUPPEN

IM Modul STO ist ein **Peer-Mentoring** („Einführung in das Studium") eingerichtet. „Ältere" Studierende aus dem Master-Bereich werden in Gruppen individuelle Lösungen erarbeiten; für Sie. Hierzu treffen Sie sich ab übernächster Woche im Mentoring. Do und Freitag dieser Woche sind die ersten Vorlesungen.

Organisieren Sie Lerngruppen, nicht nur zur Prüfungsvorbereitung. Dies wird Ihnen helfen, effektiv und effizient den Lernstoff zu bewältigen. **Alle großen Programmieraufgaben (4 in diesem Semester)** müssen Sie im Team bearbeiten (**zu Zweit! – Pair-Programming**).

3.2 EVALUATION

Alle Pflicht-Veranstaltungen des Bachelorstudiengangs Informatik werden evaluiert, das heißt, durch einen 2-seitigen Fragebogen wird Ihre Meinung zu der jeweiligen Veranstaltung (Vorlesung und Übung getrennt) erhoben. Die Anonymisierung ist absolut sichergestellt.