

»Siehst du, Momo«, sagte er dann zum Beispiel, »es ist so: Manchmal hat man eine sehr lange Straße vor sich. Man denkt, die ist so schrecklich lang; das kann man niemals schaffen, denkt man.«

Er blickte eine Weile schweigend vor sich hin, dann fuhr er fort: »Und dann fängt man an, sich zu eilen. Und man eilt sich immer mehr. Jedesmal, wenn man aufblickt, sieht man, daß es gar nicht weniger wird, was noch vor einem liegt. Und man strengt sich noch mehr an, man kriegt es mit der Angst, und zum Schluß ist man ganz außer Puste und kann nicht mehr. Und die Straße liegt immer noch vor einem. So darf man es nicht machen.«

Er dachte einige Zeit nach. Dann sprach er weiter: »Man darf nie an die ganze Straße auf einmal denken, verstehst Du? Man muß nur an den nächsten Schritt denken, an den nächsten Atemzug, an den nächsten Besenstrich. Und immer wieder nur an den nächsten.«

Wieder hielt er inne und überlegte, ehe er hinzufügte: »Dann macht es Freude; das ist wichtig, dann macht man seine Sache gut. Und so soll es sein.«

Und abermals nach einer langen Pause fuhr er fort: »Auf einmal merkt man, daß man Schritt für die Schritt die ganze Straße gemacht hat. Man hat gar nicht gemerkt wie, und man ist nicht außer Puste.«

Er nickte vor sich hin und sagte abschließend: »Das ist wichtig.«

Michael Ende, Momo

wie wird in diesem Buch Mathematik erklärt und wie lernt man Mathematik?

Man kann über Mathematik erzählen, man kann Mathematik als eine Sammlung von Rezepten für Anwender (ohne Beweise) präsentieren und man kann Mathematik als das exakte, logisch fundierte Gebäude aufbauen, das sie ihrer Natur nach ist. In diesem Buch finden Sie Darstellungen aus all diesen Blickwinkeln. Allerdings habe ich mich bemüht, einen gewissen Standard an mathematischer Exaktheit zu Grunde zu legen und stets beizubehalten. Ohne diesen Standard können Sie sehr schwer richtige und falsche Ergebnisse Ihrer Anwendungen voneinander unterscheiden. Ohne ein mathematisches Verständnis der Theorie fällt es viel schwerer, diese zu beherrschen und korrekt einzusetzen. Und ohne dieses Verständnis sind Sie nicht in der Lage, auch in neuen Situationen und bei nicht im Lehrbuch beschriebenen Problemen zu Lösungen zu kommen.

Alle wichtigen mathematischen Konzepte, mit denen Sie im Laufe Ihres Studiums konfrontiert werden, sollen logisch einwandfrei und genau definiert werden. Vor den exakten Definitionen jedoch möchte ich Ihnen immer, wenn es irgend möglich ist, erklären

- welche Bedeutung das entsprechende Konzept in der Informatik hat,
- wie es dazu kam, dass der entsprechende Begriff »erfunden« wurde, welche Leute daran beteiligt waren, warum sie es gemacht haben und wann das eigentlich war,
- und welche anschaulichen Vorstellungen einer – oft sehr abstrakten – Definition zu Grunde liegen.

Alle Sätze, die ganze Theorie wird an Beispielen erklärt und verdeutlicht und abgesehen von einigen Ausnahmen – fast alle in der Stochastik – werden auch alle Sätze bewiesen. Nur mit Hilfe der Beispiele können Sie es lernen, mit den allgemeinen Begriffen und Sätzen der Theorie umzugehen und sie erfolgreich auf weitere, neue Beispiele anzuwenden.

Und das bedeutet:

Sie müssen diese Beispiele (nach)rechnen und müssen die Bearbeitung dieser Beispiele verstehen. Bei Unklarheiten fragen Sie Ihre Dozentinnen und Dozenten, besprechen Sie sich untereinander. Sie müssen und sollten lernen, im Team zu arbeiten – Sie vermeiden es so, sich in Sackgassen zu verirren, mehr Leute haben mehr Ideen und Sie lernen es, anderen Leuten zuzuhören und anderen Leuten etwas zu erklären. Das wichtigste aber ist:

Bearbeiten Sie die Übungsaufgaben.

Man lernt Mathematik nur, wenn man sie selber »macht«. Ehrenwort! Das gilt übrigens auch für Informatik, für Physik, für alle Ingenieurwissenschaften, wahrscheinlich für alle Lernprozesse überhaupt. Sie erwerben den Führerschein nicht dadurch, dass Sie lesen, wie man fährt, sondern nur dadurch, dass Sie selber fahren. Erst in der Anwendung des gelesenen oder gehörten Stoffes auf Probleme, bei der Sie nicht mehr an der Hand geführt werden, werden Ihnen Leistungsfähigkeit und Anwendungsbereiche der Theorie klar, die Sie vorher »abgenickt« haben. Meistens versteht man sie erst in solch einer Situation.

Diese Einsicht bedeutet zunächst ein wenig mehr Arbeit, aber andererseits garantiert Sie Ihnen eine sehr erfreuliche Antwort auf eine Ihrer dringendsten Fragen.

wie kann man dieses dicke Buch zur effektiven Klausurvorbereitung nutzen?

Pro Semester umfasst dieses Buch ungefähr 30 Seiten mit Übungsaufgaben. Rechnen Sie diese Übungsaufgaben. Der Index zu diesem Buch hilft Ihnen bei der Suche nach der benötigten Theorie. Wenn Sie überhaupt nicht weiterkommen bzw. wenn Sie Ihre eigenen Lösungen überprüfen wollen, gibt es Hilfe: Auf meiner Homepage bzw. auf der entsprechenden Seite des Verlages finden Sie umfangreiche Materialien zu diesem Buch, unter anderem habe ich dort Lösungsvorschläge für sämtliche Übungsaufgaben abgestellt¹. Arbeiten Sie auch hier mit anderen Studentinnen und Studenten zusammen. Erst in der Klausur selber verlangen wir unbedingte Einzelarbeit.

Und verfahren Sie genauso mit den Übungsaufgaben, die Ihnen Ihre Dozentin oder Ihr Dozent in der Vorlesung stellt. Sie sind wirklich das Herzstück Ihres Lernfortschritts.

welche Mathematik brauchen Informatikerinnen und Informatiker?

Auf diese Frage gibt es mehrere mögliche Antworten. Deprimierende und weniger deprimierende. Die Vielfalt der Möglichkeiten zeigt sich unter anderem darin, dass jedes Lehrbuch zu diesem Thema mit einer anderen Stoffauswahl, der man zuweilen noch die Herkunft aus dem klassischen ingenieurwissenschaftlichen Bereich anmerkt, auf diese Frage antwortet.

¹ Beachten Sie dazu den Hinweis zu Beginn dieses Buches.

Die Antwort, die Sie (und mich und meine Kolleginnen und Kollegen) am ratlosesten macht, heißt: Sie brauchen alles, Sie brauchen die ganze Mathematik, Analysis, Algebra, Zahlentheorie, Statistik und so weiter, und so weiter. Denn in diesen Gebieten wird überall Informatik eingesetzt, bzw. es gibt für jedes dieser Gebiete Programme im wissenschaftlichen und kommerziellen Bereich, die Mathematik aus diesen Bereichen benutzen und in denen mathematische Formeln aus diesen Bereichen programmiert werden.

Aber: Nach dieser Logik müssten Informatikerinnen und Informatiker über alles Mögliche in der Welt Bescheid wissen: über Physik, über Materialwirtschaft, über Gehaltsabrechnungen, über Biologie, über Fantasy-Welten, über Lohnsteuergesetze, über englische, deutsche, japanische, französische und chinesische Sprachen und vieles mehr. Denn in all diesen Bereichen gibt es Anwendungen aus der Informatik, aus all diesen Bereichen sind Regeln und Verhaltensweisen in Programmen formalisiert worden.

Sie merken: Das kann man von keinem Informatiker verlangen. Die Lösung muss woanders liegen. Sie müssen eine der wichtigsten Fähigkeiten erwerben, die ein Informatiker zur erfolgreichen Arbeit braucht: die Fähigkeit, den Anwender, den Kunden zu zwingen, seine Spezifikationen exakt und verständlich anzugeben. Der Experte aus der Anwenderwelt muss alles, was wichtig für die Software-Modellierung ist, so bereitstellen, dass Sie es korrekt einarbeiten können. Betrachten Sie dazu folgenden Beispieldialog zwischen einer Person S, die die Spezifikation für ein Programm macht und einer Person R, die realisieren soll:

S: »Please write a program to search for an element in a table!«

R: »What kind of a table? A list? An array? A tree? Are the elements sorted? Do they have a key?«

S: »I don't want to consider these implementation issues. That is your job. «

R: »But what should be done if the sought element is not in the table?«

S: »Sorry?«

Frei übersetzt:

S: »Bitte schreiben Sie ein Programm, das ein Element in einer Tabelle sucht!«

R: »Welcher Art ist diese Tabelle? Eine Liste? Ein Array? Ein Baum? Sind die Elemente in dieser Tabelle sortiert? Haben sie einen Schlüssel?«

S: »Mit diesen Fragen der Implementierung möchte ich mich nicht beschäftigen. Das ist Ihre Angelegenheit.«

R: »Wie sollte das Programm reagieren, wenn das gesuchte Element nicht in der Tabelle vorhanden ist?«

S: »Wie bitte?«

Dieses Beispiel ist dem englischsprachigen Buch »Understanding Formal Methods« [Monin] entnommen und wird über dort über mehrere(!) Kapitel hinweg diskutiert. Unter anderem dient es dazu, klar zu machen, wie wichtig eine genaue Spezifikation der Eingangs- und Ausgangsbedingungen eines Programms ist.

Fassen wir zusammen: Sie müssen *nicht* die gesamte Mathematik kennen, aber Sie müssen sich anwenderspezifisches Expertenwissen so bereitstellen lassen können, dass Sie Eingangs- und Ausgangszustände Ihres Programms und die jeweiligen Verarbeitungsregeln genauestens beschreiben können. *Dafür* braucht man Mathematik, unter anderem Mengenlehre, Aussagenlogik und Prädikatenlogik.

Wir können uns jetzt einer Antwort auf die oben gestellte Frage »Welche Mathematik brauchen Informatikerinnen und Informatiker?« nähern, die nicht mehr so deprimierend ist, wie unser erster Versuch. Grundlage für die Formulierung dieser Antwort ist das Prinzip:

Wir sagen, dass eine mathematische Theorie in der Informatik gebraucht wird, falls sie echter Bestandteil eines Gebiets der Informatik ist.

Das ist natürlich nur die Illusion einer Definition, trotzdem war dieses Prinzip für mich eine gute Richtschnur für die Stoffauswahl für dieses Buch und für die Antwort auf unsere Frage. Ich habe in jedem Kapitel, in jedem Abschnitt versucht, zu erklären, inwieweit der jeweils behandelte Stoff zur Informatik gehört und wo er gebraucht wird.

Inhalt und Aufbau dieses Buches

Das Buch behandelt die folgenden Themen:

Im ersten Teil werden die bereits erwähnten Grundlagen der Aussagen- und Prädikatenlogik und der Mengenlehre behandelt. Alles, was hier steht, brauchen Sie andauernd bei Ihrer Tätigkeit als Informatikerin oder Informatiker.

Mehrere Kapitel in diesem Buch beschäftigen sich damit, Ihnen eine gute Vorstellung von der Welt der Zahlen zu geben, dazu gehört ein möglichst großes Wissen von der mathematischen Welt der Zahlen einerseits und andererseits eine klare Vorstellung von dem kleinen Ausschnitt aus dieser Welt, den Sie in einem Computerprogramm gleich welcher Art realisieren können.

Ich beginne mit dieser Darstellung im zweiten Teil, in ihm geht es um natürliche Zahlen und das Prinzip der Induktion, um ganze Zahlen und um rationale Zahlen. Und gleichzeitig werden hier algebraische Strukturen zu ihrer Beschreibung besprochen, nämlich Gruppen, Ringe und Körper.

Ehe wir diese Betrachtungen mit der Diskussion der reellen und der komplexen Zahlen und die Mandelbrotmenge abschließen, erledigen wir auf dem Weg dahin noch etwas anderes: Die Mathematik der relationalen Datenbanken, der Codierung und der Verschlüsselung. Das geschieht im dritten Teil, hier geht es um Äquivalenzrelationen, Endliche Gruppen und Körper und um Zahlentheorie.

Der vierte Teil bespricht die reellen Zahlen und die komplexen Zahlen und ihre algebraische Abgeschlossenheit. Außerdem zeige ich Ihnen, wie man das berühmte »Apfelmännchen« erzeugen kann.

Diese ersten vier Teile bilden den Stoff einer Mathematikvorlesung für Informatikstudenten des ersten Semesters. Im zweiten Semester geht es wieder mehr um diskrete Mathematik, hierzu gehört der fünfte Teil dieses Buchs, in dem die Boolesche Algebra besprochen wird.

Desgleichen umfasst der Stoff dieses zweiten Semesters die Graphentheorie, hier werden Verbindungsprobleme diskutiert, es werden kürzeste, längste, billigste und teuerste Verbindungen gesucht, Brücken überquert und Heiratschancen geprüft. Insbesondere

lernen Sie äußerst wichtige Werkzeuge für die schnelle Navigation in komplexen Datenstrukturen kennen, wie sie beispielsweise in jeder Datenbank verwendet werden. Sie werden zur Lösung dieser Probleme Algorithmen konstruieren und deren Laufzeit abschätzen. All das geschieht im sechsten Teil.

Die nächsten drei Teile beschäftigen sich mit der Stochastik (Wahrscheinlichkeitsrechnung und Statistik), typischerweise der Stoff des dritten Semesters in der Mathematikausbildung von Informatikern. Der siebte Teil beschäftigt sich mit der deskriptiven Statistik, der achte Teil mit der Wahrscheinlichkeitsrechnung und der neunte Teil hat die schließende Statistik zum Inhalt. Die Stochastik nimmt aus zwei Gründen eine besondere Stellung in diesem Buch ein:

- Zunächst scheint die Stochastik nicht in gleichem Maße ein »echter« Bestandteil der Informatik zu sein wie die anderen, bisher aufgeführten Gebiete. Allerdings erweist sich diese Einschätzung beispielsweise dann als falsch, wenn man sich mit dem hoch interessanten Gebiet des Data-Mining beschäftigt, wo statistische Methoden ein ganz zentraler Bestandteil sind.
- In den Kapiteln über Wahrscheinlichkeitsrechnung und schließende Statistik sah ich mich immer wieder gezwungen, mathematische Sätze zu benutzen, die ich nicht beweisen konnte, weil uns die dazu nötige Theorie fehlte. Das betrifft zu einem großen Teil fortgeschrittene Kenntnisse aus der Analysis. Zu solch einem Vorgehen war ich in keinem anderen Bereich dieses Buches veranlasst. Ich weiß, dass mich ein derartiges Vorgehen weit mehr beunruhigt als Sie – aber glauben Sie mir: je mehr Sie unbewiesene Sätze als »Rezepte« einsetzen, desto schwerer fällt es Ihnen, einen »Riecher«, ein Gefühl für richtige und falsche Ergebnisse Ihrer Anwendungen zu entwickeln.²

Damit kommen wir zum letzten Teil dieses Buches, es ist ein Anhang zur Analysis. Dieser Teil soll Sie an Dinge erinnern, die Sie aus der Schule wissen. Das Register hilft Ihnen dabei, dieses Wissen schnell wieder aufzufrischen. Sie brauchen diesen Anhang in diesem Buch bei drei Gelegenheiten:

- Bei der Diskussion der reellen Zahlen (Kapitel 9).
- Bei den Untersuchungen der Laufzeit von Algorithmen, wo Ihnen der Unterschied von logarithmischem, polynomialem und exponentiellem Wachstum klar werden muss (Kapitel 18).
- Und – immer wieder – in den Stochastikkapiteln 20 bis 24.

Programme, Programme, Programme

Dieses Buch ist voller Software, um den Informatikern unter Ihnen zu zeigen, wie man die Mathematik und die Algorithmen programmieren kann. Einige Programme brauchen wir auch zur Durchführung von numerischen Berechnungen, die für den Taschenrechner viel zu umfangreich oder kompliziert sind. Fast alles ist in Standard C++ geschrieben, nur für graphische Darstellungen wie z.B. die Zeichnung der Mandelbrot-Menge habe ich entscheidend die Entwicklungsumgebung von Microsoft .NET benutzt. Wichtige Programmteile sind im Text abgedruckt, die vollständigen Programme finden Sie wieder

² Ich habe Sie schon zu Beginn dieses Vorworts auf die anderen Nachteile dieser Strategie hingewiesen.

auf meiner Homepage bzw. auf der entsprechenden Seite des Verlages. Beachten Sie dazu wieder den Hinweis zu Beginn dieses Buches.

Außerdem habe ich in den Stochastikkapiteln viel mit Excel gearbeitet, einige Arbeitsblätter habe ich Ihnen ebenfalls ins Netz gestellt, wer sich mit Excel auskennt, wird mit ihnen leicht weiter arbeiten können. Alle anderen unter Ihnen, die Wahrscheinlichkeitstheorie und Statistik lernen wollen und noch keine Erfahrung mit Excel haben, sollten ein wenig Zeit investieren, um sich ein paar Grundlagen anzueignen. Es lohnt sich auf jeden Fall. Natürlich gibt es auch andere Software, mit der man in diesem Zusammenhang arbeiten kann, die genauso gut ist oder besser oder schlecht oder nur anders. Aber irgendeine Unterstützung zur Berechnung von Beispielen und statistischen Funktionen ist nützlich und hilft bei der Bearbeitung der Übungsaufgaben. Und Übungsaufgaben ... aber das sagte ich ja bereits.

Brauchen Informatikerinnen, brauchen Informatiker Mathematik?

»Informatiker brauchen Mathematik, die Informatik braucht Mathematik.« Das scheint selbstverständlich zu sein, an allen Hochschulen, Fachhochschulen und sonstigen Ausbildungsstätten, an denen Informatik gelehrt und gelernt wird, gehört Mathematik mit zum Studienprogramm, sie ist – teilweise bis in die letzten, das Studium abschließenden Semester hinein – ein fester und oft schwieriger Begleiter beim Erlernen der Informatik. Warum ist das so? Mathematik ist bis zum heutigen Tage von entscheidender Bedeutung bei allen Fragen, die die Hardware, den technischen Bereich der Informatik betreffen. Aber die Mathematik spielt auch im Bereich der Software eine sehr wichtige Rolle. Es gibt dafür mehrere Gründe – Gründe, die mit einer tiefer liegenden Beziehung zwischen diesen beiden Wissenschaften zu tun haben und die auch erklären, warum sich die Informatik zu einem großen Teil aus dem Schoß der Mathematik heraus entwickelt hat.

Am Anfang waren die Mathematiker bei der Bearbeitung von Problemen der Informatik meistens unter sich, denn sie waren an Rechnern, an Computern – »computer« ist das englische Wort für »Rechner« – interessiert, an Maschinen, die sie beim Rechnen unterstützten. Im Laufe der Zeit entwickelte sich die Informatik aber immer mehr zu einer eigenständigen Wissenschaft, die auch alle möglichen anderen Probleme bearbeiten können sollte. Diese Entwicklung ist in den letzten Jahren immer schneller und umfassender vorangeschritten und betrifft alle Bereiche des menschlichen Lebens. Damit haben wir scheinbar eine Wegentwicklung von der Mathematik, aber in Wirklichkeit ist das Gegenteil der Fall. Ich gebe Ihnen ein einfaches Beispiel: Wenn Sie sich in einem kleinen Notizbuch die Namen, Adressen und Telefonnummern Ihrer Freunde und Geschäftspartner anlegen, wird fast jeder Eintrag sein eigenes »Format« haben: Ein Eintrag hat keine Telefonnummer, ein anderer hat zwei oder drei, einer steht nur mit seinem Vornamen da, ein anderer Eintrag ist achtungsgebietend mit Titel und Firmenbezeichnung notiert, Sie können nicht immer richtig sortieren, weil der Platz aufgebraucht ist, manchmal sortieren Sie kategorial (z. B. Freunde im Inland, Freunde im Ausland), manchmal alphabetisch, manchmal »nach Wichtigkeit«. Sobald Sie diese Aufgabe von einem Rechner durchführen lassen wollen, ist Schluss mit dieser Individualbehandlung. Sie müssen eine allgemei-

ne Form finden, die alle Sonderfälle mit erfasst. Und wenn Sie Ihre Adresseinträge nach verschiedenen Kriterien (Name, Telefonnummer, Adresse usw.) sortieren wollen, wollen Sie dafür nur ein Programm schreiben, nicht pro Kriterium eins. Überhaupt wollen Sie in Ihrem Leben nur ein einziges Sortierprogramm schreiben, das für alle möglichen Dinge wie Zahlen, Personen, Texte, Worte, Buchstaben, Artikel eines Warenhauses und was da sonst noch sortiert werden kann, gleichermaßen funktioniert. Sie müssen allgemeine Strukturen erkennen können, konkrete Dinge wie Personen und Verkaufsartikel auf diese allgemeinen Strukturen hin abstrahieren können und diese Abstraktionen verarbeiten können bzw. in der objektorientierten Programmierung mit den richtigen Fähigkeiten zum aktiven Handeln versehen können. Auch diese Formalisierung des Verhaltens von abstrakten Strukturen ist eine zutiefst mathematische Tätigkeit. Es ist eng verwandt mit dem Entwurf von Algorithmen.

Diese Fähigkeit zur Abstraktion und zum Arbeiten mit Abstraktionen lernen Sie nirgendwo besser als in der Mathematik, Sie lernen sie nur in der Mathematik. Die Mathematik beschäftigt sich – sehr vereinfacht gesprochen – mit Objekten, die auf numerische und räumliche Eigenschaften hin reduziert sind. Diese Abstraktionen sind so stark, die Abstraktionsebene ist so hoch, dass viele Menschen oft denken, Mathematik habe mit der Realität überhaupt nichts zu tun. Das stimmt ganz offensichtlich nicht, dazu wird Mathematik in allen möglichen Bereichen des Lebens viel zu oft erfolgreich angewandt. Das geschieht aber auf der Grundlage einer so starken Abstraktion, einer so hohen Abstraktionsebene, dass ein und dieselbe Mathematik in den unterschiedlichsten realen Situationen angewendet werden kann. Welche andere Wissenschaft kann beispielsweise in so verschiedenartigen Anwendungsgebieten wie bei Marketingaktionen, in der Quantenphysik, bei politischen Meinungsumfragen, bei mathematischen Primzahlanalysen und in der medizinischen Forschung erfolgreich zum Einsatz kommen wie die Statistik? (Davon werden Sie übrigens im letzten Teil dieses Buches einen guten Eindruck bekommen). Und das geht nur,

- weil die entsprechenden Fragestellungen auf einem Niveau formuliert werden, das genügend von den für das Problem unwichtigen Einzelheiten und Eigenschaften abstrahiert
- und weil die »verbleibenden« abstrakten Strukturen mit der nötigen Tiefe und Qualität untersucht werden.

Diese beiden Fähigkeiten, die Fähigkeit zur guten, zur richtigen Abstraktion und zum effektiven Arbeiten und Formalisieren auf der einmal gewonnenen Abstraktionsebene, sind entscheidende Qualifikationen für die gute Softwareentwicklerin und für den guten Softwareentwickler.

- Sie wollen, dass Ihre Programme so oft wie möglich und in so unterschiedlichen Situationen wie möglich wieder verwendet werden können. Das bedeutet, Sie müssen Ihr Abstraktionsniveau so hoch wie möglich anlegen.
- Sie wollen, dass spätere Änderungen in den konkreten Anwendungen – auch in einer späten Phase der Implementierung – leicht eingebaut werden können. Das bedeutet: Ihre Abstraktionen müssen von hoher Qualität sein³.
- Und Sie wollen, dass Ihre Programme so schnell, so effektiv wie möglich arbeiten. Das bedeutet, Sie müssen mit den abstrakten Strukturen gut und souverän umgehen können.

³ Was das bedeutet, lernen Sie in einer Vorlesung über Software-Engineering und Entwurfsmuster.

Wieso, höre ich Sie völlig zu Recht fragen, wieso lernen wir diese Formen der Abstraktion und der Verarbeitung von abstrakten Strukturen nicht gleich an dem Ort und an den Beispielen, die ganz direkt aus der Informatik kommen?

Die Antwort darauf besteht aus mehreren Teilen:

- Gerade zu Beginn Ihres Studiums sind Sie in der Informatikausbildung viel zu sehr damit beschäftigt, erst einmal die Sprachen wie C++, Java oder C# zu lernen, in denen Sie später diese Abstraktionen modellieren und verarbeiten werden. Die Erlernung und Beherrschung dieser Sprachen ist etwas ganz anderes als der Entwurf einer guten und effektiven Softwarearchitektur – aber sie muss diesem Entwurf vorausgehen.
- Der Entwurf einer guten und effektiven Softwarearchitektur ist in hohem Maße nicht trivial, vergleichen Sie dazu eines meiner Lieblingsbücher zu diesem Thema [Mart]. So etwas können Sie nicht am Anfang Ihres Studiums lernen.
- Die Sprache der Mathematik hingegen steht Ihnen schon zu Beginn Ihres Studiums seit vielen Jahren zur Verfügung. In dieser Sprache können Sie sehr schnell leichte und auch schwierigere Probleme formulieren und bearbeiten, die die oben beschriebenen Fähigkeiten schulen und vertiefen. Sie werden sehr direkt, direkter als Ihnen zunächst bewusst ist, auf das Studium komplexer Software-Techniken und auf das Verständnis von Entwurfsmustern und anderen Abstraktionen vorbereitet.
- Und Sie müssen sowieso einige mathematische Kenntnisse erwerben, die Sie unmittelbar in Ihrer Informatikausbildung brauchen.

Ceterum Censeo – Im übrigen bin ich der Meinung

Vor einigen Jahren schrieb ich im Vorwort zu meinem Buch über Datenbanken:

»Dieses Buch wurde geschrieben im ständigen Gespräch, in der kontinuierlichen Diskussion mit ›meinen‹ Studentinnen und Studenten. Es verdankt ihnen alles: den Aufbau, die Beispiele, die Genauigkeit der Erklärungen und den gesprächsartigen, persönlichen Stil, in dem dieses Buch geschrieben wurde. In diesen Zeiten der Knappheit in den öffentlichen Kassen, der fortwährenden Umstrukturierungen im Hochschulbereich, der Einführung von Studiengebühren haben es die Studierenden wahrlich nicht leicht, ihr Studium konzentriert durchzuführen. Die allermeisten sind gezwungen, nebenher zu arbeiten, um das Studium zu finanzieren und nur wenige haben das Glück, einen ›Job‹ zu finden, der fachlich in enger Beziehung zum Studienfach steht. Umso großartiger ist es, in jedem Semester immer wieder eine große Zahl von Studentinnen und Studenten zu finden, die sich mit Interesse, Engagement und – sehr wichtig – Ausdauer an der Diskussion und Erarbeitung des Stoffes beteiligen. Manchmal scheint es, als ob die gegenwärtige Hochschulpolitik nur darauf aus ist, diese kreative Arbeitsatmosphäre immer unmöglicher zu machen. Lassen Sie uns alle dafür sorgen, dass das nicht geschieht.«

Diese Worte sind noch genauso wahr wie vor vier Jahren, die Anzahl der bürokratischen Regelungen mit Vorleistungen, Vorbedingungen und Exmatrikulationsandrohungen hat sich dagegen dramatisch erhöht und belastet das Studium mit zahlreichen, völlig fachfremden Hürden und Behinderungen. Man möchte den Reglementierern zurufen: Hört auf, solchen Schaden anzurichten und lasst uns in Ruhe arbeiten! Ich hoffe sehr, dass dieser Prozess noch umkehrbar ist.

Danke, Danke, Danke

Ich stehe in mehrfacher Hinsicht auf sehr breiten Schultern, ich verdanke sehr viel meinen großartigen Lehrern. Das betrifft sowohl die Schule als auch in besonderem Maße die Universität. Insbesondere konnte ich unter Bedingungen studieren, die sehr sparsam in Bezug auf formale Regelungen waren, dafür hoch, ja höchst engagiert in Bezug auf die Qualität der mathematischen Inhalte und der pädagogischen Konzepte. Ich wünsche Ihnen, liebe Studentinnen und Studenten, von ganzem Herzen vergleichbare Studiumsmöglichkeiten – nicht ohne Sie darauf hinzuweisen, dass die Studenten damals diese Situation auch nicht geschenkt bekommen haben. Sie haben oft hart dafür kämpfen müssen.

Viel in diesem Buch verdanke ich der Diskussion mit meinen Kollegen. Insbesondere die Stochastik habe ich mir von Grund auf von meinem Kollegen Egbert Falkenberg erklären lassen müssen. Ich habe – von manchem Schmunzeln der anderen Studenten begleitet – seine Vorlesung besucht und sein wunderschönes Skript durchgearbeitet. Wann immer ich etwas aus diesem Skript in meinem Buch verwende, habe ich es deutlich gemacht, aber der ganze Stochastikabschnitt in diesem Buch basiert insgesamt auf dieser Vorlesung. Nur die Fehler, die sind von mir. Für diese Hilfe auch noch mal hier ein großes Dankeschön.

Und dann Sie, die Studentinnen und Studenten! Mit Ihnen, in vielen Gesprächen und durch den Vergleich verschiedener Lösungswege, durch Ihre Kritik und Ihren Zuspruch ist dieses Buch in vielen Jahren entstanden, gewachsen und besser geworden. Ihre Reaktion hat mir Mut gemacht, Ihre Kritik hat mich nachdenklich gemacht, Ihre Klausuren haben mich manchmal überrascht und mir gezeigt, was ich besser machen muss. Auch Ihnen gebührt ein großer Dank. Wenn Sie dieses Buch lesen und Kritik und Verbesserungsvorschläge haben oder wenn Sie mir nur einfach Ihre Meinung sagen wollen, bitte, schreiben Sie mir. Auf meiner Homepage oder auf der Seite der Fachhochschule Frankfurt finden Sie meine Adresse. Oder schreiben Sie an den Verlag. Wir brauchen Ihre Reaktion.

Ulrich Sandten vom Verlag Vieweg+Teubner hatte die Idee zu diesem Buch. Er hat es mir vorgeschlagen und ich war sofort Feuer und Flamme. Die Realisierung dieses Buchs hat viel länger gedauert als zunächst geplant war, aber er hat mich in der gesamten Zeit großartig unterstützt, er hat mir Mut gemacht, wenn ich dabei war, ihn zu verlieren und er hat mit vielen Ideen zu Form und Inhalt dieses Buchs beigetragen. Dafür meinen allerherzlichsten Dank – ihm und dem gesamten Verlags-Team unter der Leitung von Ulrike Schmickler-Hirzebruch, die einen so tollen Job machen. Und danke an Ivonne Domnick, die meinen »nackten« Text in so eine schöne Form gebracht hat.

Ich bedanke mich ebenso bei »meiner« Fachhochschule Frankfurt (am Main), die mir im Sommersemester 2008 ein »Sabbatical« gewährt hat, ohne das dieses Buch wahrscheinlich nie fertig geworden wäre.

Carsten Biemann, ein ehemaliger Diplomand und jetzt ein guter Freund, hat das gesamte Buch sorgfältig Korrektur gelesen, hat viele Fehler gefunden, hat viel kritisiert, sehr selten gelobt und war eine große Hilfe. Danke auch ihm.

Ich widme dieses Buch »meinen« Studentinnen und Studenten, ohne die alles nichts wäre. Und natürlich meiner geliebten Marianne, die von Anfang an dagegen war.

Vorwort zur zweiten Auflage

When you are swinging, swing more!

Thelonious Monk

Drei Dinge brauchte diese neue Auflage

Zum ersten mussten natürlich einige Irrtümer, Druck- und Layoutfehler korrigiert werden. Zum zweiten hatte ich sowohl bei der Beschreibung des Ungarischen Algorithmus als auch bei der Definition der Akzeptoren und der Zurückweiser im Kapitel über die Min- und Maxterme auf einmal das Gefühl, dass ich da etwas völlig Unverständliches geschrieben hatte. Das war ein halbes Jahr nach der Erstellung der ersten Fassung, und ich war erschrocken, wie schnell sich das subjektive Gefühl von Verständlichkeit ändern kann. Ich habe diese Abschnitte also noch einmal völlig neu geschrieben und habe sie vor Drucklegung auch noch einmal von mehreren Leuten gegenlesen lassen. Diese neuen Versionen sind auch von den Studentinnen und Studenten freundlicher aufgenommen worden. Ein weiteres Kapitel, das große, für mich unerwartete Schwierigkeiten gemacht hat, war das Kapitel über die Komplexität von Algorithmen. Ich habe zum Verständnis der Begriffe P und NP noch einen weiteren kleinen Abschnitt hinzugefügt. Zum dritten (seufz) wollte der Ruf nach der linearen Algebra nicht verstummen. Ich habe mich ergeben und zwei Kapitel zu diesem Thema verfasst, die die wichtigsten Begriffe aus diesem Gebiet enthalten. Ich versuche dort auch, Ihnen zu vermitteln, warum dieser Teil der Mathematik für die Informatik eine große Relevanz haben kann. Es sind die neuen Kapitel 11 und 12.

Ein weiteres Mal: Danke

Ich muss mich bei allen, die mir so vielfältige und wichtige Rückmeldungen gegeben haben, bedanken. Dazu gehören Meldungen von Fehlern oder Ungereimtheiten, dazu gehören sehr freundliche, aufmunternde und lobende Worte aber auch sehr kritische. Ich bedanke mich bei meinem Verlag, bei Frau Ulrike Schmickler-Hirzebruch und Herrn Ulrich Sandten für die phantastische Betreuung, ich bedanke mich bei Ivonne Domnick für die tolle graphische Gestaltung und bei meinen Kollegen Dieter Hackenbracht und Jens Lorenz für ihre gründliche Kritik und ihre überaus freundlichen Worte.

Ich widme auch diese Auflage meinen Studentinnen und Studenten, die alle zu ihrer Entstehung so viel beigetragen haben. Und natürlich meiner geliebten Marianne, die übrigens (darum beneide ich sie) Thelonious Monk persönlich gekannt hat.

Frankfurt, im August 2011 Matthias Schubert

<http://www.springer.com/978-3-8348-1848-5>

Mathematik für Informatiker

Ausführlich erklärt mit vielen Programmbeispielen und
Aufgaben

Schubert, M.

2012, 844 S. 118 Abb., Softcover

ISBN: 978-3-8348-1848-5