

**Christian Horn, Immo O. Kerner (Hrsg.): Lehr- und Übungsbuch Informatik.**

**Band 3: Praktische Informatik.**

Fachbuchverlag Leipzig im Carl Hanser Verlag München Wien, 1997, ISBN 3-446-18699-9, 405 Seiten, Preis ?.

## Praktische Informatik für ungeübte Studenten

Ein weiteres Lehrbuch der Informatik. Diesmal für Studenten der allgemeinen und der Wirtschaftsinformatik an den Deutschen Fachhochschulen gedacht. Die Praktische Informatik ist Band 3 eines vierbändigen Werkes. Die anderen drei Bände sind den folgenden Themen gewidmet:

Band 1: Grundlagen und Überblick

Band 2: Theorie der Informatik (in Vorbereitung)

Band 4: Technische Informatik und Systemgestaltung (für Herbst 1997 geplant)

Am ganzen Werk sind 30 Autoren beteiligt. Am Band 3, den wir zu lesen bekamen, sind es sieben, zwischen Dresden und Stralsund zu Hause. Im Vorwort erfahren wir: „Die Praktische Informatik umfasst im weitesten Sinne alle Massnahmen, die den Computer ‘zum Laufen’ bringen, ihn also zu nutzbringender Arbeit veranlassen. Im Vordergrund steht die Entwicklung und Implementierung von Programmiersprachen.“

In der Praxis wird kaum mit der Entwicklung von Programmiersprachen dem Rechner das Laufen beigebracht, zum Glück auch im Buch nicht; es behandelt die Entwicklung und Implementierung von Programmen, unter Zuhilfenahme von verschiedenen Programmiersprachen. Die Themen sind:

- Programmierung im Kleinen mit PASCAL
- Software-Engineering
- Datenbankverwaltung mit SQL
- Prolog-Programmierung
- Funktionale Programmierung mit SCHEME
- Objektorientierte Analyse und Programmierung
- Grafische Datenverarbeitung
- Informatik und Gesellschaft

Die Reihe ist als Lehr- und Übungsbuch konzipiert, d.h. der Leser findet Aufgaben, an denen er das erworbene Wissen versuchen kann umzusetzen.

### Zur Struktur

Die Struktur ist sinnvoll, einzig beim objektorientierten Kapitel ergibt sich ein Bruch. Die objektorientierte Analyse ist ein Thema des Software Engineering und verdiente, neben und gleichwertig mit der strukturierten Analyse behandelt zu werden. Das objektorientierte Programmieren dürfte ruhig ein eigenes Kapitel wie Prolog oder SCHEME haben, müsste aber auch anhand einer konkreten Programmiersprache behandelt werden – welcher, müssen zum Glück nicht wir entscheiden. Derart wäre das Buch ausgewogener und würde dem heutigen Stand der Dinge mehr Rechnung tragen.

Ziemlich unverhofft kommt das letzte Kapitel. Wir finden die ethischen Fragen im Zusammenhang mit der Informatik sehr wichtig, sie werden in Lehrbüchern zu wenig oder gar nicht diskutiert. Die Aufnahme in dieses Werk ist lobenswert, nur ... ob der Band Praktische Informatik dazu der richtige Ort ist?

An den Seitenrändern findet man Stichwörter und Piktogramme, eine gute Orientierungshilfe. Die Bedeutung der Piktogramme muss man allerdings zuerst einmal entziffern; vielleicht sind sie im Band 1 erklärt.

Zu jedem Kapitel gibt es am Ende ein Verzeichnis mit weiterführender Literatur; zum Teil wird auch im Text darauf verwiesen. Leider werden die Titel, auch die im Text nicht erwähnten, ohne Kommentar aufgeführt. Die Anordnung ist bei fünf Kapiteln alphabetisch nach Autoren, bei den andern (Kapitel 2, 6 und 8) ohne ersichtliche Logik. Mindestens eine Referenz (/2.57/ V-Modell) fehlt.

Am Ende des Bandes findet man ein zumeist befriedigendes Sachwortverzeichnis. Allerdings fehlen – unserer Meinung nach – einige Hinweise, während andere völlig überflüssig sind. So gibt es z.B. für „Booch“ zwar einen Eintrag für die Seite 319, wo man die Referenz (!) /6.9/ auf die Angabe im Literaturverzeichnis auf Seite 320 findet, aber keinen für die Seite 265, auf welcher die Booch-Methode in groben Zügen vorgestellt wird. Andererseits gibt es einen Eintrag für „revision“, der uns auf Seite 101 zu einer Übungsaufgabe über ein Modul führt, welches ca. 25 Seiten weiter vorne steht und zufälligerweise „revision“ heisst.

## Zur Sprache und Begriffsbildung

Rein geographisch ist die Entfernung von der Schweiz nach Dresden oder Stralsund sicher grösser als von Bayern oder Niedersachsen. Ob dies auch für den Sprachgebrauch in der Informatik zutrifft, wissen wir nicht. Für unsere Augen und Ohren sind aber Begriffe wie „Nutzeroberflächen“ (Seite 18), oder „Entprozedurieren“ (Seite 25) sehr ungewohnt. Während beim ersten leicht erahnt werden kann, dass es sich um die „Benutzeroberfläche“ handelt (übrigens sind beide Begriffe an sich unsinnig; Gegenstand der Betrachtung ist keineswegs unsere Oberfläche, wenn wir als Nutzer oder Benutzer fungieren), hilft beim zweiten auch der englische (?) Begriff „deprocedurering“ nicht weiter.

Unser englisches Sprachgefühl hilft jedoch zu erkennen, dass „reengineering“ keinesfalls Deutsch für das Englische „reverse engineering“ ist (Seite 46). Man kann sehr gut reengineeren, ohne den Reverse-Gang einzulegen, wenn der Entwurf (Design) aktuell dokumentiert vorliegt. Wenn! Angesichts der Tatsache, dass es um Praktische Informatik geht, ist vielleicht das Gleichsetzen doch nicht so falsch ...

Wenn wir schon beim Deutsch-Englischen Dialog sind: Löblicherweise wird an den meisten Stellen – leider nicht überall – neben dem deutschen Begriff auch der englische angegeben; dies erleichtert die Orientierung für den Studenten, der sich auch in die englische Fachliteratur verirrt.

Gelegentlich werden Begriffe ohne Erklärung verwendet. So wird z.B. das Halteproblem erwähnt (Seiten 375 und 378), aber es wird weder gesagt was es ist noch wo man etwas darüber nachlesen könnte. Vermutlich wird es in Band 2 behandelt.

## Zum Inhalt

Ein dickes Lob verdient die Tatsache, dass ein Unterkapitel mit elf Seiten der Qualitätssicherung gewidmet ist. Es wird zwar die ganze Breite des Qualitätsmanagements vorgestellt, behandelt werden jedoch nur die Themen Metriken und Prüfen. Die Abhandlung über Metriken ist (Praktische Informatik!) adäquat, diejenige des Prüfens jedoch nur, wenn in einem anderen Band (Theoretische Informatik?) das Thema vertieft wird.

Wie sollen Absolventen der Deutschen Fachhochschulen in der Praxis vernünftig in Entwicklungsteams mitarbeiten können, wenn sie über Reviews knapp eine Seite gelesen haben und während des Studiums (vermutlich) nie eines erlebt haben? Wie sollen sie fehlerarme Programme abliefern, wenn sie gute drei Seiten über das Testen gelesen, in der Vorlesung eine Viertelstunde zugehört haben und während des ganzen Studiums (vermutlich) nie eine Testspezifikation oder einen Testbericht abliefern mussten? Sollten wir zu diesen Aussagen Widerspruch der Autoren ernten, würde es uns sehr freuen.

Validierung, Reviews und Verifizierung werden als Varianten der statischen Prüfung aufgeführt. Wie die Validierung praktisch gemacht werden soll, bleibt unklar. Reviews sind kurz vorgestellt. Die Verifizierung wird als formale Beweisführung definiert.

Diese drei Dinge sind nicht orthogonal und, ausser Review, umfassen sie nicht nur statische Prüfungen. Reviews und Tests (dynamische Prüfung) sind Verfahren, die sowohl für Validierung als auch für Verifizierung eingesetzt werden können. Der Unterschied liegt in der Vorgabe, gegen die geprüft wird. Sind es bei der Verifizierung die spezifizierten Anforderungen, so müssen dafür bei der Validierung die eher nebulösen Bedürfnisse der Anwender erhalten. Dies ist zumindest das Verständnis dieser Begriffe in der Qualitätswelt.

Faktisch falsch ist, dass Verifizierung „der mathematische Nachweis der Korrektheit eines Programms“ ist. Diese Definition trifft nur auf die sogenannte formale Verifizierung zu. Reviews und Tests sind durchaus Verfahren zur Verifizierung der Erfüllung von Anforderungen. Im Buch sind Reviews in die Nähe der Validierung gerückt, was den Eindruck erweckt, dass Reviews ein Verfahren für die Validierung sind (was explizit nicht behauptet wird).

Die Programmiersprachen werden gut, aber mit unterschiedlichem Tiefgang behandelt. So wird PASCAL im ersten Kapitel nicht bis in alle Details vorgestellt, während im fünften Kapitel über die funktionale Programmierung sogar die Tastenkombinationen für den in PC SCHEME verwendeten Editor aufgeführt sind! Die verwendeten Beispiele sind durchwegs einfach nachvollziehbar und doch illustrativ.

Zeit- und leserkreisgerecht ist die Angabe einer Web-Site im Vorwort, an der man über die Buchreihe, bekannte Fehler und Lösungen der Kontrollaufgaben informiert wird; die Adresse führt (nicht selbstverständlich) zum Ziel. Zum besprochenen Band findet man noch keine Fehler und Lösungen, den Zugang zu den Angaben zum Band 1 muss man erst finden (über den Link mit dem Namen des ersten Herausgebers, nicht jedoch des zweiten).

## Zusammenfassung

Das gesamte Werk verspricht ein wertvolles Lehr- und Lernmittel für Fachschulstudenten zu werden. Im vorliegenden Band finden sie eine gute Übersicht über Programmiersprachen; alle heute praxisrelevanten Programmier-Paradigmen sind vertreten. Zum Selbststudium würden wir das Werk nicht empfehlen; das tun auch die Autoren nicht.

31. Dezember 1997

Karol Frühauf + Ruedi Schild, *INFOGEM AG*, 5401 Baden