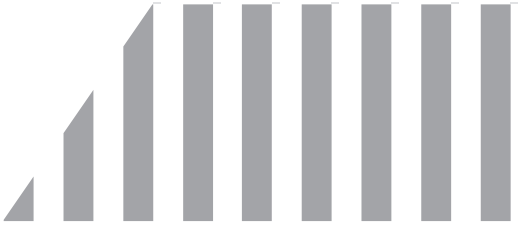
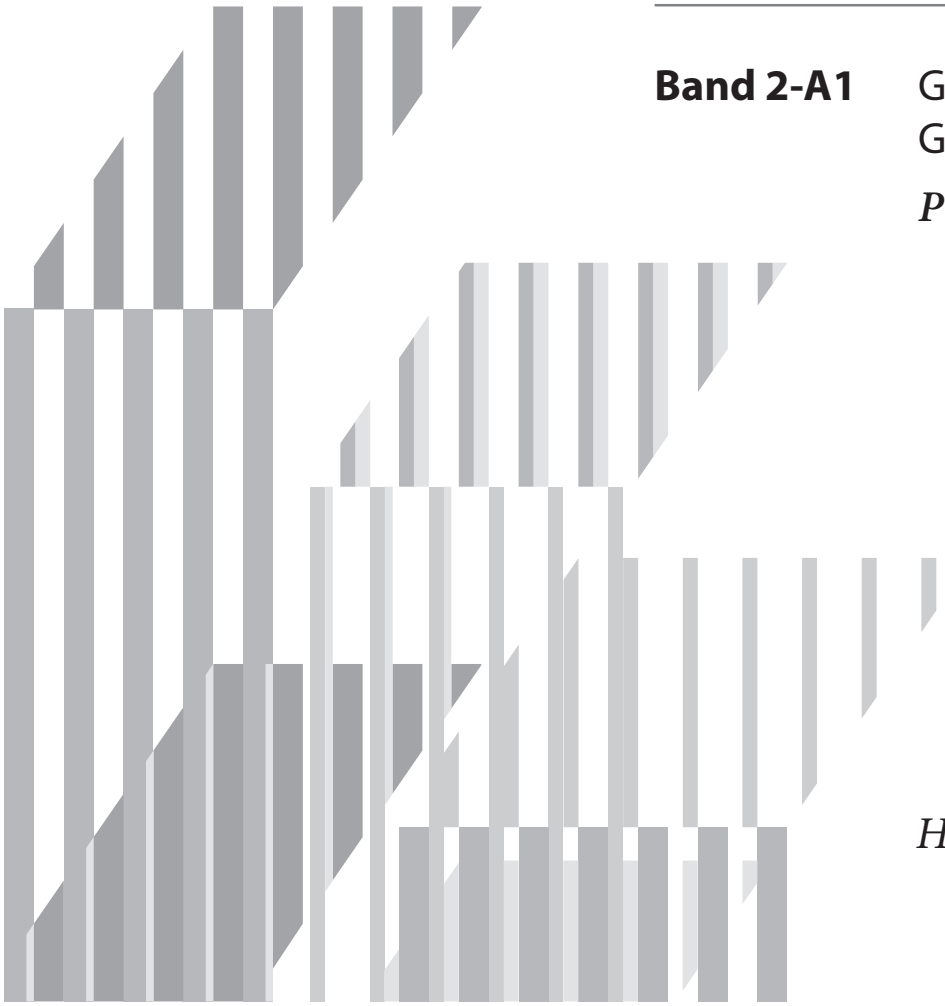
Band 2-A1



MODERNE STRUKTURIERTE PROGRAMMIERUNG

Objektorientiert und algorithmisch
ENTWERFEN | IMPLEMENTIEREN | TESTEN

2. Auflage



Band 2-A1 Grundlagen
Gruppen
Praxis

Horst van Bremen

Bibliografische Information der Deutschen Nationalbibliothek:
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.dnb.de abrufbar.

Gestaltung	Katharina Schmitt
Titel- und Einbandgrafik	Monika Sylvia Gomm † 1971
Englisch-Korrektorat	David Roseveare
Verlag	BoD · Books on Demand GmbH, Überseering 33, 22297 Hamburg, bod@bod.de
Druck	Libri Plureos GmbH, Friedensallee 273, 22763 Hamburg
Version / Datum	Version 2.1.1 vom 1.7.2026
Literaturverzeichnis	siehe Kapitel „7.1 Literaturverzeichnis“ auf Seite 139 f
Rechtliches	siehe Kapitel „7.2 Rechtliche Hinweise“ auf Seite 141 ff
ISBN des Ringbuchs	978-3-6963-1357-9

Über den Autor

Horst van Bremen
Diplom 1972 an der TU Darmstadt – Elektrotechnik und Informatik
39 Berufsjahre in der IT, davon 18 als Freiberufler

IT-Systemarchitekt
Datenbankexperte

Programmiererfahrung seit 1969
Strukturierte Programmierung nach Jackson seit 1974
Freier Autor seit 2011

Vorwort zur ersten A1-Ausgabe

Der Band 2-A1 der ersten Staffel dieser Buchreihe versammelt alle Programmbeispiele, die auf der Basis der JSP/MSP-Entwürfe im Band 1-A1 vom Autor geschrieben wurden, sowie Testprotokolle. Das ist zumindest unüblich. Normalerweise enthalten Bücher über Programmierung zwar Code-Beispiele in Form kurzer Abschnitte, die englisch Snippets – also Schnipsel oder Bruchstück – genannt werden, aber den Code ganzer Programme – und erst recht Tests hiervon – hat der Autor noch in keinem Fachbuch gesehen. Im vorliegenden Fall gibt es aber für das ungewöhnliche Vorgehen gute Gründe:

- Anfänger tun sich häufig schwer damit, etwas, das sie theoretisch verstanden haben, in ein funktionierendes Programm umzusetzen. Die dafür erforderliche Erfahrung braucht ihre Zeit, um sich zu entwickeln. Um so besser und schneller geht es, wenn Beispiele vorliegen, an denen sich die eigenen Entwürfe orientieren können. Die Download-Option auf www.jsp-msp.de erspart das Abtippen.
- In Jahrzehnten des Ausprobierens und des Scheiterns bildeten sich Kriterien dafür, wie eine professionelle Programmierung aussehen kann, nicht muss. Ein Beispiel: In den allerersten ALGOL-Programmen des Autors hießen die Variablen A1, A2, A3 und so weiter, was diesen Code von vornherein nahezu unverständlich machte. Wie wichtig es ist, passende Namen zu wählen, stellte sich erst nach und nach heraus. Hier möchten der Band 2-A1 und seine Folgebände stilbildend sein.
- Große Unsicherheiten bestehen anfangs auch bei der Modularisierung, sowie bei der OO-Programmierung, ganz besonders aber bei der Zuordnung von Code-Snippets zu Methoden. Wie man hier geschickt vorgeht, das lehrt auch JSP/MSP nicht. Hinsichtlich dieses Aspekts wird in den Praxis-Bänden ab A1 angestrebt, zumindest gute Lösungen zu zeigen, ohne den Anspruch auf Vorbildlichkeit zu erheben.
- Vollständig ungewöhnlich ist es, JSP/MSP-Entwürfe in OO-Sprachen zu realisieren. Die strukturierte, datenbasierte Methode von Michael A. Jackson wurde zwar primär für die algorithmische Programmierung konzipiert, eignet sich aber auch für OO-Implementierungen. Das Zusammenspiel „echter" OO-Klassen mit den neu eingeführten Process Control-Klassen, die sich den JSP-typischen Baumstrukturen verdanken, kann anhand der hier vorgelegten Beispiele studiert werden. In den Folgebänden wird dieses Vorgehen angewandt, um letztlich auch komplexe Programmstrukturen in plausiblen OO-Code umzusetzen.
- Die Aufteilung der Buchreihe in Paare von Methoden- und Praxis-Bänden ist nicht nur dem Umfang der Bände geschuldet, sondern sie dient hauptsächlich dazu, die Entwürfe und ihre Umsetzungen nebeneinanderlegen zu können. Durch die konsequente Übernahme der Operations- und Kontroll-Bezeichnungen in die Programme und durch die ausführliche Kommentierung kann die Ableitung des Codes aus den Struktogrammen gut nachvollzogen werden. Die Test-Abschnitte unterstützen das Verständnis.
- Code-externe Kommentare, Anmerkungen und teilweise recht detaillierte Darstellungen dienen dazu, den Implementierungen eine eigene Semantik zu geben. Auch gut kommentierter Code ist selten wirklich selbsterklärend. Die Zusatztexte sollen die verbleibende Verständnislücke schließen helfen.
- Um „den Wald" zu schonen, wurden Redundanzen nach Kräften vermieden. Wiederverwendeter Code wird nicht wiederholt, sondern es wird auf dessen erstes Auftreten verwiesen.

Es ist dem Autor bewusst, dass es vermutlich ebensoviele Meinungen über die „richtige" Programmierung wie Programmierer(innen) gibt. Daher ist es möglich, dass die hier gezeigten Implementierungen auf heftige Kritik oder gar auf völlige Ablehnung stoßen. Ganz sicher gibt es auch an den nicht immer konsequent umgesetzten Namenskonventionen und anderen Formalia dies und jenes zu bemängeln. Insofern hofft der Autor auf die Gnade derjenigen, die es damit ganz besonders streng halten. Allen anderen mögen die Kapitel des Bands 2-A1 hoffentlich von großem Nutzen sein.

Lemgo, den 14.2.2025

Übersichtsverzeichnis Band 2-A1

Inhaltsverzeichnis Band 2-A1