

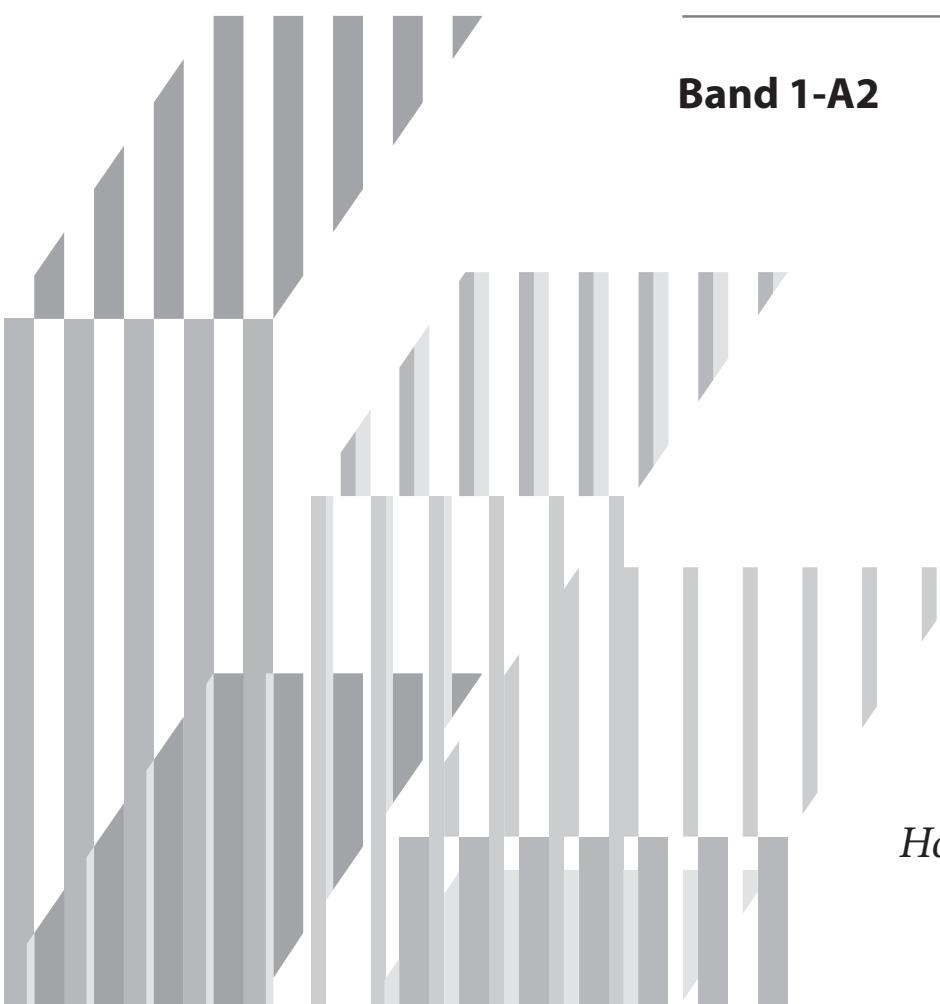
MODERNE STRUKTURIERTE PROGRAMMIERUNG

Objektorientiert und algorithmisch
ENTWERFEN | IMPLEMENTIEREN | TESTEN

Band 1-A2

Ranggruppen
Link-Listen
Methode

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.

© 2025 Dipl.-Ing. Horst van Bremen

Gestaltung	Katharina Schmitt
Titel- und Einbandgrafik	Monika Sylvia Gomm † 1971
Englisch-Korrektorat	David Roseveare
Verlag	HMG®Verlags-GmbH, Neue Strasse 74, 32657 Lemgo Mail-Adresse: office@hmg-verlag.de Website: www.hmg-verlag.de
Druck	Libri Plureos GmbH, Friedensallee 273, 22763 Hamburg
Version / Datum	Version 1.1.3 vom 14.2.2025
Literaturverzeichnis	siehe Kapitel „18.1 Literaturverzeichnis“ auf Seite 573 ff.
Rechtliches	siehe Kapitel „18.2 Rechtliche Hinweise“ auf Seite 575 ff.
ISBN des Paperback-Buchs	978-3-910566-31-6
ISBN der Paperback-Buchreihe	978-3-910566-00-2

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 A2-Ausgabe

Ranggruppen und Link-Listen – abstrakter geht es kaum. Was unter Laien als eine dieser vielen seltsamen Sprachverirrungen gelten mag, für welche die Informatik berühmt-berüchtigt ist, besitzt dennoch eine hohe praktische Relevanz. Telefonbücher oder Online-Verzeichnisse, Wörterbücher, Inhaltsverzeichnisse, steuerliche Auswertungen, Bundesligatabellen, Bibliotheken-Bestandsregister, Anlagevermögen-Aufstellungen und zahllose weitere Auflistungen basieren auf einem Spektrum von manuellen bis hochautomatisierten Techniken, bei denen es um – eventuell hierarchisch geschachtelte – Reihenfolgen oder Rankings geht.

Sie alle lassen sich mit JSP-Struktogrammen beschreiben und – datentechnisch aufbereitet – in vielfältiger Weise gestalten und verwerten. Allerdings sind diese nützlichen und allgegenwärtigen Listen meistens völlig frei von jeglichem intellektuellen und ästhetischen Reiz, also eine knochentrockene Angelegenheit, die ein Autor weder seinen Leserinnen und Lesern noch sich selbst als interessante Thematik schmackhaft zu machen vermag. Die Fallstudie zum fiktiven Internationalen Schulsportwettbewerb im hier vorliegenden Teil A2 des Bands 1 bildet dagegen sowohl hinsichtlich der Anschaulichkeit als auch der Komplexität eine Ausnahme in dieser Ödnis – so steht es jedenfalls zu hoffen. Letztlich entscheiden das natürlich Sie als Leserin oder Leser.

Die Texte und Grafiken der Fallstudie sind nicht als lineare Abfolge von Lektionen gedacht. Das wäre nicht nur extrem langweilig, sondern würde auch dem Lernziel nicht gerecht, das Schwierigkeitsniveau A2 bei der Beherrschung von JSP/MSP zu erreichen. Das bedeutet, dass hier die Grundlagen für eine selbständige Anwendung der Methode und für deren praktische Umsetzung gelegt werden sollen. Repräsentative Anwendungsfälle für JSP/MSP, beispielsweise in der kommerziellen Programmierung, sind nämlich allenfalls so komplex wie die Fallstudie. Daher behandeln die folgenden Kapitel

- die Abbildung des „Daten-Rohmaterials“ in die aufgabenspezifischen Struktogramme,
- deren Umsetzung in die Programmstrukturen,
- die Verifikation dieser Strukturen anhand von Beispielfällen,
- eine inkrementelle algorithmische Implementierung und die zugehörigen Tests,
- die Transformation der JSP/MSP-Entwürfe mit dem Ziel der OO-Implementierung,
- die Anpassung der Basismaschine an die Eigenschaften der verwendeten Java-Klassen,
- den Entwurf einer flexiblen Datei-Zugriffsschicht für Java-Programme und
- das Thema „Test“ aus mehreren Perspektiven.

Auch das sind abstrakt erscheinende Themen, aber ihre Umsetzung anhand der Fallstudie ermöglicht es, die methodische Stringenz mit der Anschaulichkeit der Wettbewerbsaspekte zu verbinden. Final geht es darum, das Potential von JSP/MSP auch bei komplexen Aufgaben zu demonstrieren und das Vorgehen zu deren Lösung anhand überschaubarer Beispiele nachvollziehbar darzustellen. Bei dem Schritt, die hier gewonnenen Erkenntnisse in Ihre eigene berufliche Praxis umzusetzen, wünsche ich, der Autor, Ihnen ein gutes Händchen.

Lemgo, den 14.2.2025

Übersichtsverzeichnis Band 1-A2

9 Ranggruppenverarbeitung I	191
10 Link-Listen-Verarbeitung	237
11 Ranggruppenverarbeitung II	291
12 Test I	311
13 OO-Vorbereitungen für EvaluateSportsDS	359
14 File Services	429
15 Test II	441
16 Test III	499
17 Ranggruppenverarbeitung III	515
18 Anhang	573



Inhaltsverzeichnis Band 1-A2

9 Ranggruppenverarbeitung I	191
9.1 Der Internationale Schulsportwettbewerb	192
9.1.1 Vorläufige Eingabe-Datenstruktur	192
9.1.2 Ausgabe-Format	197
9.1.3 Datenflussdiagramm	200
9.1.4 Vorläufige Ausgabe-Datenstruktur	201
9.2 Erzeugen der reduzierten Ausgabe	204
9.2.1 Finales Sports Dataset-Struktogramm	204
9.2.2 Vereinfachte Programmstruktur ohne Operationen	206
9.2.3 Umwandlung der Iterationskontrollen I2 bis I6	207
9.2.4 Definition und Zuordnung von Operationen	208
9.2.5 Durchlaufanalyse anhand der vereinfachten Programmstruktur	209
9.2.6 Implementierung der vereinfachten Programmstruktur	220
9.2.7 Implementierung in COBOL II	221
9.2.8 Implementierung in REXX	223
9.3 Hinzunahme der Schul-Daten	225
9.3.1 Zeitreise zurück in die 60er Jahre	225
9.3.2 $10^3 - 10^6 - 10^9 - 10^{12} - 10^{15} - 10^{18} - 10^{21} - 10^{24} - \dots$	228
9.3.3 Technische Minimalkonfiguration 2022	229
9.3.4 Schul-Daten in einer zweiten Datei	230
10 Link-Listen-Verarbeitung	237
10.1 Die Sort-Methode	237
10.2 Die Datenbank-Methode	238
10.3 Die Häufchen-Methode	238
10.4 Die 2D-Array-Methode	241
10.5 Die Stapel-Methode	243
10.6 Die Linked List-Methode	245
10.6.1 Hinzunahme der zweiten Punktsumme (PS2)	245
10.6.2 Hinzunahme der dritten Punktsumme (PS3)	247
10.6.3 Hinzunahme der vierten Punktsumme (PS4)	249
10.6.4 Hinzunahme der fünften Punktsumme (PS5) etcetera	251
10.6.5 Grenzen des Wachstums / zyklische Bereinigung	251
10.7 Umsetzung der Linked List-Methode	252
10.7.1 Neutrale Modellierung der Link-Liste	252
10.7.2 Haben wir uns verirrt?	257
10.7.3 Aufbauen der Link-Liste	257
10.7.4 Ist dieser Algorithmus performant?	260
10.7.5 Fertigstellung des Eingabe-Struktogramms	262
10.7.6 Ableiten des Ausgabe-Struktogramms	263
10.7.7 Programmstruktur ohne Operationen	265
10.7.8 Definition der Link-Liste	266
10.7.9 Liste der Operationen	267
10.7.10 Präzisierung der Kontrollen	268
10.7.11 Programmstruktur mit Operationen	268

Band 1-A2

10.7.12 Umspeichern in die leere Listenhälfte	274
10.8 Durchlaufanalyse	276
10.8.1 Erstellen des ersten Punktsummen-Eintrags	276
10.8.2 Hinzunahme der zweiten Punktsumme (PS2)	277
10.8.3 Hinzunahme der dritten Punktsumme (PS3)	279
10.8.4 Hinzunahme der vierten Punktsumme (PS4) etcetera	284
10.9 Operationen und Kontrollen der Link-Listen-Verarbeitung	288
11 Ranggruppenverarbeitung II	291
11.1 Schreiben und Lesen der Link-Listen	291
11.2 Finales School-Sports-Result-Struktogramm	293
11.3 Korrespondenzen	297
11.4 Programmstruktur ohne Operationen	299
11.5 Daten für buildLinkedList()	301
11.5.1 Teilnehmerdaten für die Schüler	301
11.5.2 Teilnehmerdaten für die Schulen	301
11.5.3 Link-Listen	301
11.6 Fertigstellung der Operationsliste	302
11.7 Finale Programmstruktur mit Operationen	304
11.8 Implementierung in C	306
11.9 Operationen und Kontrollen von EvaluateSportsDS (C)	306
12 Test I	311
12.1 Der wahre Test ist die Produktion	311
12.1.1 Hans im Pech	312
12.1.2 T.O.O. S.M.A.R.T.	312
12.2 Was lernen wir daraus?	316
12.2.1 Elementares	316
12.2.2 Schwieriges	318
12.3 Testen eines fremden Programms	321
12.3.1 Testmethoden	322
12.3.2 Black-Box-Test des Programms	325
12.3.3 Code-Analyse	326
12.3.4 Breakpoint Debugging / Tracing	328
12.4 Ein- und Aussichten	329
12.4.1 Wieviel Testen ist genug?	329
12.4.2 Das traditionelle Programmierdilemma	330
12.4.3 Testen auf JSP-Basis	331
12.4.4 Testen hybrider Programme	333
12.4.5 Driver-Finale	334
12.5 Test von EvaluateSportsDS in C	337
12.5.1 Einbau der Trace-Aufrufe	338
12.5.2 Test der Ranggruppenverarbeitung	339
12.5.3 Test der Schulstammdatenverarbeitung	343
12.5.4 Test der Link-Listen-Verarbeitung	344
12.5.5 Test mit einem umfangreichen Bestand	349
12.5.6 Bewertung der Tests	354
12.6 Laufergebnisse	354

13	OO-Vorbereitungen für EvaluateSportsDS	359
13.1	Das Schwierigste zuerst	359
13.2	Rangschlüssel als Objekte	360
13.2.1	Entity Relationship-Modellierung kontra Rang-Sicht	361
13.2.2	OO-Modellierung der Rang-Sicht	363
13.2.3	Generalisierbarkeit des OO-Modells	367
13.2.4	Sonderfall: Nur ein Rang = eine Gruppe	370
13.3	Nutzung der Java-Link-Listen-Container	371
13.4	Aufbauen von Link-Listen	372
13.4.1	1. Eintrag anlegen	372
13.4.2	2. Eintrag hinzufügen	373
13.4.3	3. und 4. Eintrag hinzufügen	373
13.5	Link-Listen-Polymorphie	375
13.5.1	Oberklasse Contestant	376
13.5.2	Klasse Pupil	376
13.5.3	Klasse School	377
13.5.4	Anlegen der Link-Listen	377
13.5.5	Einfügen von Einträgen	378
13.5.6	Auswertung mit Hilfe der Oberklasse	379
13.5.7	Anlegen von Schüler-Einträgen	379
13.5.8	Voranstellen und positionsrichtiges Einfügen von Einträgen	380
13.5.9	Löschen obsolet gewordener Einträge	382
13.5.10	Fazit	382
13.6	Austausch der Link-Listen-Basismaschine	383
13.6.1	Erste Auswirkungen auf die Operationsliste	383
13.6.2	Strukturdefinitionen	385
13.6.3	Anpassungen an die neue Basismaschine	386
13.6.4	Neue / abgewandelte Operationen	392
13.7	Klassen identifizieren	394
13.7.1	Ablösen der Operation 22	395
13.7.2	Struktogramm-Änderung	396
13.7.3	Zwischenspeicherung der Zählergebnisse	398
13.7.4	Daten & Operationen ==> Klassen?	400
13.7.5	Link-Listen: Was ist Klasse, was Objekt, was Parameter?	402
13.7.6	Vorläufige Klassenliste	403
13.8	Zuständigkeiten festlegen	404
13.9	Operationen den Klassen zuordnen	409
13.10	Übergang zu den Klassendiagrammen	414
13.11	Wie sind die Klassendiagramme zu lesen?	420
13.11.1	Top-Diagramm	420
13.11.2	Erstes Teildiagramm: Einbindung der File Service-Klassen	422
13.11.3	Zweites Teildiagramm: Einbindung der Rangschlüsselklassen	423
13.11.4	Drittes Teildiagramm: Contest-Klassen und deren Nutzer	423
13.12	Implementierung in Java	423
13.13	Operationen und Kontrollen von EvaluateSportsDS (Java)	423
14	File Services	429
14.1	Numerische Datei-Identifikationen	431
14.2	Erster Vorschlag	431

Band 1-A2

14.3 Zweiter Vorschlag	434
15 Test II	441
15.1 The Second-System Effect	441
15.2 Regressionstest	442
15.3 Test der File Services	445
15.3.1 Testleitfaden	446
15.3.2 Test von <code>setupInFiles()</code> mit <code>readFiles = 0</code>	449
15.3.3 Test von <code>setupInFiles()</code> mit <code>readFiles = 2</code>	459
15.3.4 Test von <code>open()</code> für den Sports Dataset	461
15.3.5 Test von <code>open()</code> für den School Sport Results Dataset	467
15.3.6 Fazit	474
15.4 Programmierte Pfadprotokollierung	475
15.4.1 Pfadprotokollierung mit Dateiausgabe	475
15.4.2 Pfadprotokollierung mit relationaler Ausgabe	483
15.4.3 Tracing & Breakpoint Debugging: Risiken und Nebenwirkungen	485
15.4.4 Pfadprotokollierung mit Ausgabe in einen Trace-Speicher	488
16 Test III	499
16.1 The Whole and the Parts	500
16.2 Das Zeiterfassungssystem und EXAMINE	501
16.3 (Integrations-)Test von EvaluateSportsDS	502
17 Ranggruppenverarbeitung III	515
17.1 Einfaches Java-E-Mail-Programm	516
17.2 Datenstrukturen und Ableiten der Programmstruktur	518
17.2.1 Vorläufiges Eingabe-Struktogramm	518
17.2.2 Korrespondenzen zwischen Ein- und Ausgabe	518
17.2.3 Neue Korrespondenztypen: verzögert / erweitert	522
17.2.4 Finale Eingabe-Datenstruktur mit Korrespondenzen	523
17.2.5 Finale Ausgabe-Datenstruktur mit Korrespondenzen	525
17.2.6 Programmstruktur ohne Operationen	529
17.3 Sprachabhängige Aufbereitung	532
17.4 Technisches Format der E-Mail-Daten	535
17.5 Programmstruktur mit Operationen	539
17.5.1 Strukturdefinitionen	539
17.5.2 Definition und Zuordnung von Operationen	542
17.5.3 Top-Struktogramm mit Operationen	545
17.5.4 Ausgelagerte Struktogramme mit Operationen	547
17.6 Durchlaufanalyse	549
17.7 Was hat die Durchlaufanalyse gebracht?	571
17.8 Implementierung in Java	572
18 Anhang	573
18.1 Literaturverzeichnis	573
18.2 Rechtliche Hinweise	575
18.2.1 Geschützte Bezeichnungen	575

Inhalt

18.2.2 Haftungsausschluss	582
18.2.3 Zitate	583
18.2.4 Firmen, Personen und Namen	583
18.2.5 Bild- und Grafiknachweis	584
18.2.6 Programme	584
18.3 Versionsgeschichte zu 1.1.3	584

