

Übersichtsverzeichnis Band 1

1	Einführung	1
2	JSP-Basismethode	19
3	Algorithmische Implementierungsvorbereitungen	53
4	Objektorientierte Implementierungsvorbereitungen	79
5	Blick zurück und voraus	93
6	Durchlaufanalyse	111
7	Gruppenverarbeitung I	123
8	Gruppenverarbeitung II	173
9	Ranggruppenverarbeitung I	187
10	Link-Listen-Verarbeitung	233
11	Ranggruppenverarbeitung II	287
12	Test I	307
13	OO-Vorbereitungen für EvaluateSportsDS	355
14	File Services	425
15	Test II	437
16	Test III	495
17	Ranggruppenverarbeitung III	511
18	Anhang	569

Inhaltsverzeichnis Band 1

1	Einführung	1
1.1	JSP-Literatur	3
1.2	Wo stehen wir heute?	4
1.3	Programme mit MSP konstruieren	5
1.4	... und die Folgen	7
1.5	Die Buchreihe	8
1.6	In eigener Sache	10
1.7	Last but not least	12
2	JSP-Basismethode	19
2.1	Vorbemerkungen	19
2.2	Das erste JSP-Beispiel	20
2.3	Der intuitive Ansatz	22
2.4	Konstruieren anstatt Erfinden	23
2.5	Strukturen der Ein- und Ausgabedaten identifizieren	24
2.5.1	Vom konkreten Eingabebeispiel zur abstrakten Eingabestruktur	24
2.5.2	I = Iteration	25
2.5.3	S = Selection (Auswahl)	26
2.5.4	Blöcke ohne S oder I	26
2.5.5	Heikle Ratschläge	26
2.5.6	Sonderfall Leerzeile	28
2.5.7	Unbemerkte Implikationen	28
2.5.8	Vom konkreten Ausgabebeispiel zur abstrakten Ausgabestruktur	29
2.5.9	Inkompatible Strukturblöcke	31
2.6	Korrespondenzen zwischen den Datenstrukturen ermitteln	31
2.7	Programmstruktur ohne Operationen ableiten	32
2.8	Programmstruktur mit Operationen ergänzen	34
2.9	Nachbemerkungen zu den Operationen	43
2.10	Kontrollen überprüfen und präzisieren	44
2.11	Ausgabe-Lösungsalternative	46
3	Algorithmische Implementierungsvorbereitungen	53
3.1	Flussdiagramm erstellen	54
3.2	Schematische Logik oder Pseudocode schreiben	57
3.3	Jacksons Schematische Logik	58
3.4	BDL-Pseudocode	63
3.4.1	Exkurs zum Hintergrund von BDL	63
3.4.2	Die BDL-Syntax	69
3.5	Anmerkungen zum Pseudocode	74
4	Objektorientierte Implementierungsvorbereitungen	79
4.1	JSP objektorientiert?	79
4.2	Klassen und Objekte bestimmen	81

4.3	Bewertung der Ergebnisse	86
4.4	Namenskonventionen	86
5	Blick zurück und voraus	93
5.1	Und das soll alles sein?	93
5.2	Wie geht es weiter?	95
5.2.1	Durchlaufanalyse als Sicherheitsnetz	96
5.2.2	Der Klassiker: (Rang-)Gruppenverarbeitung	96
5.2.3	Fallstudie zur Ranggruppenverarbeitung	97
5.3	Mischen: Das unerschöpfliche Thema	99
5.3.1	Misch-Standardfälle	99
5.3.2	Mischen mit Plausibilitätsprüfung	99
5.3.3	Mischen mit unterschiedlich tiefen Rangschlüsseln	100
5.3.4	Mischen von drei und mehr Beständen	101
5.3.5	Multidimensionales Mischen	101
5.3.6	Abgleich von Bilddateien mit ihren Kopien	101
5.3.7	Exkurs: Generierung von Change-Kommandos	102
5.3.8	Abgleich von Dateien in beliebig tief gestaffelten Verzeichnissen	102
5.3.9	Ende und neuer Anfang	104
5.4	Backtracking und Programminversion	104
5.4.1	Backtracking in drei Lektionen	105
5.4.2	Programminversion	105
5.5	Zusammenfassung	106
5.6	JSP/MSP und relationale Datenbanktechniken	107
5.7	Geschafft!	109
6	Durchlaufanalyse	111
6.1	Durchlaufanalyse = Ablaufsimulation	111
6.2	Zusammenfassung	119
6.3	Problembehandlung	119
6.4	Testkontrolle	120
7	Gruppenverarbeitung I	123
7.1	Sonett-Typ-Erkennung	124
7.2	Sonett-Eingabestruktogramme	124
7.3	Generalisierung kontra Spezialisierung	128
7.4	Überlegungen zu den Kontrollen	132
7.5	Finale Struktogramme	135
7.6	Operationen	139
7.7	Italienisches Sonett: Beispiel und Struktur	141
7.8	Durchlaufanalyse	142
7.8.1	Sonderfall: Nach 6 Absätzen folgen weitere Zeilen	149
7.8.2	Sonderfall: Dateiende nach dem 1. Absatz	156
7.8.3	Sonderfall: Leerzeile(n) + Dateiende nach dem 1. Absatz	156
7.8.4	Vorsicht Falle!	159
7.8.5	Schlussbemerkungen zur Durchlaufanalyse	161
7.9	Pseudocode	162
7.10	Implementierungs-Varianten in COBOL II und REXX	164

7.10.1	COBOL II	164
7.10.2	REXX	166
7.11	Implementierung in C und Java	170
8	Gruppenverarbeitung II	173
8.1	Struktogramme von Ein- und Ausgabe	175
8.2	Programmstruktur, Operationen und Kontrollen	176
8.3	Programmstruktur mit Operationen	177
8.4	Durchlaufanalyse	178
8.4.1	Initialphase	178
8.4.2	Folgesatzverarbeitungen	179
8.4.3	Verarbeitung der ersten Duplikate	181
8.4.4	Verarbeitung des Gruppenendes und des Folgesatzes	183
8.4.5	Schlussphase	184
8.5	Implementierung in C und Java	184
8.6	Der geplatzte Traum	185
9	Ranggruppenverarbeitung I	187
9.1	Der Internationale Schulsportwettbewerb	188
9.1.1	Vorläufige Eingabe-Datenstruktur	188
9.1.2	Ausgabe-Format	193
9.1.3	Datenflussdiagramm	196
9.1.4	Vorläufige Ausgabe-Datenstruktur	197
9.2	Erzeugen der reduzierten Ausgabe	200
9.2.1	Finales Sports Dataset-Struktogramm	200
9.2.2	Vereinfachte Programmstruktur ohne Operationen	202
9.2.3	Umwandlung der Iterationskontrollen I2 bis I6	203
9.2.4	Definition und Zuordnung von Operationen	204
9.2.5	Durchlaufanalyse anhand der vereinfachten Programmstruktur	205
9.2.6	Implementierung der vereinfachten Programmstruktur	216
9.2.7	Implementierung in COBOL II	217
9.2.8	Implementierung in REXX	219
9.3	Hinzunahme der Schul-Daten	221
9.3.1	Zeitreise zurück in die 60er Jahre	221
9.3.2	$10^3 - 10^6 - 10^9 - 10^{12} - 10^{15} - 10^{18} - 10^{21} - 10^{24} - \dots$	224
9.3.3	Technische Minimalkonfiguration 2022	225
9.3.4	Schul-Daten in einer zweiten Datei	226
10	Link-Listen-Verarbeitung	233
10.1	Die Sort-Methode	233
10.2	Die Datenbank-Methode	234
10.3	Die Häufchen-Methode	234
10.4	Die 2D-Array-Methode	237
10.5	Die Stapel-Methode	239
10.6	Die Linked List-Methode	241
10.6.1	Hinzunahme der zweiten Punktsumme (PS2)	241
10.6.2	Hinzunahme der dritten Punktsumme (PS3)	243
10.6.3	Hinzunahme der vierten Punktsumme (PS4)	245

10.6.4	Hinzunahme der fünften Punktsumme (PS5) etcetera	247
10.6.5	Grenzen des Wachstums / zyklische Bereinigung	247
10.7	Umsetzung der Linked List-Methode	248
10.7.1	Neutrale Modellierung der Link-Liste	248
10.7.2	Haben wir uns verirrt?	253
10.7.3	Aufbauen der Link-Liste	253
10.7.4	Ist dieser Algorithmus performant?	256
10.7.5	Fertigstellung des Eingabe-Struktogramms	258
10.7.6	Ableiten des Ausgabe-Struktogramms	259
10.7.7	Programmstruktur ohne Operationen	261
10.7.8	Definition der Link-Liste	262
10.7.9	Liste der Operationen	263
10.7.10	Präzisierung der Kontrollen	264
10.7.11	Programmstruktur mit Operationen	264
10.7.12	Umspeichern in die leere Listenhälfte	270
10.8	Durchlaufanalyse	272
10.8.1	Erstellen des ersten Punktsummen-Eintrags	272
10.8.2	Hinzunahme der zweiten Punktsumme (PS2)	273
10.8.3	Hinzunahme der dritten Punktsumme (PS3)	275
10.8.4	Hinzunahme der vierten Punktsumme (PS4) etcetera	280
10.9	Operationen und Kontrollen der Link-Listen-Verarbeitung	284
11	Ranggruppenverarbeitung II	287
11.1	Schreiben und Lesen der Link-Listen	287
11.2	Finales School-Sports-Result-Struktogramm	289
11.3	Korrespondenzen	293
11.4	Programmstruktur ohne Operationen	295
11.5	Daten für buildLinkedList()	297
11.5.1	Teilnehmerdaten für die Schüler	297
11.5.2	Teilnehmerdaten für die Schulen	297
11.5.3	Link-Listen	297
11.6	Fertigstellung der Operationsliste	298
11.7	Finale Programmstruktur mit Operationen	300
11.8	Implementierung in C	302
11.9	Operationen und Kontrollen von EvaluateSportsDS (C)	302
12	Test I	307
12.1	Der wahre Test ist die Produktion	307
12.1.1	Hans im Pech	308
12.1.2	T.O.O. S.M.A.R.T.	308
12.2	Was lernen wir daraus?	312
12.2.1	Elementares	312
12.2.2	Schwieriges	314
12.3	Testen eines fremden Programms	317
12.3.1	Testmethoden	318
12.3.2	Black-Box-Test des Programms	321
12.3.3	Code-Analyse	322
12.3.4	Breakpoint Debugging / Tracing	324
12.4	Ein- und Aussichten	325

12.4.1	Wieviel Testen ist genug?	325
12.4.2	Das traditionelle Programmierdilemma	326
12.4.3	Testen auf JSP-Basis	327
12.4.4	Testen hybrider Programme	329
12.4.5	Driver-Finale	330
12.5	Test von EvaluateSportsDS in C	333
12.5.1	Einbau der Trace-Aufrufe	334
12.5.2	Test der Ranggruppenverarbeitung	335
12.5.3	Test der Schulstammdatenverarbeitung	339
12.5.4	Test der Link-Listen-Verarbeitung	340
12.5.5	Test mit einem umfangreichen Bestand	345
12.5.6	Bewertung der Tests	350
12.6	Laufergebnisse	350
13	OO-Vorbereitungen für EvaluateSportsDS	355
13.1	Das Schwierigste zuerst	355
13.2	Rangschlüssel als Objekte	356
13.2.1	Entity Relationship-Modellierung kontra Rang-Sicht	357
13.2.2	OO-Modellierung der Rang-Sicht	359
13.2.3	Generalisierbarkeit des OO-Modells	363
13.2.4	Sonderfall: Nur ein Rang = eine Gruppe	366
13.3	Nutzung der Java-Link-Listen-Container	367
13.4	Aufbauen von Link-Listen	368
13.4.1	1. Eintrag anlegen	368
13.4.2	2. Eintrag hinzufügen	369
13.4.3	3. und 4. Eintrag hinzufügen	369
13.5	Link-Listen-Polymorphie	371
13.5.1	Oberklasse Contestant	372
13.5.2	Klasse Pupil	372
13.5.3	Klasse School	373
13.5.4	Anlegen der Link-Listen	373
13.5.5	Einfügen von Einträgen	374
13.5.6	Auswertung mit Hilfe der Oberklasse	375
13.5.7	Anlegen von Schüler-Einträgen	375
13.5.8	Voranstellen und positionsrichtiges Einfügen von Einträgen	376
13.5.9	Löschen obsolet gewordener Einträge	378
13.5.10	Fazit	378
13.6	Austausch der Link-Listen-Basismaschine	379
13.6.1	Erste Auswirkungen auf die Operationsliste	379
13.6.2	Strukturdefinitionen	381
13.6.3	Anpassungen an die neue Basismaschine	382
13.6.4	Neue / abgewandelte Operationen	388
13.7	Klassen identifizieren	390
13.7.1	Ablösen der Operation 22	391
13.7.2	Struktogramm-Änderung	392
13.7.3	Zwischenspeicherung der Zählergebnisse	394
13.7.4	Daten & Operationen ==> Klassen?	396
13.7.5	Link-Listen: Was ist Klasse, was Objekt, was Parameter?	398
13.7.6	Vorläufige Klassenliste	399
13.8	Zuständigkeiten festlegen	400

13.9	Operationen den Klassen zuordnen	405
13.10	Übergang zu den Klassendiagrammen	410
13.11	Wie sind die Klassendiagramme zu lesen?	416
13.11.1	Top-Diagramm	416
13.11.2	Erstes Teildiagramm: Einbindung der File Service-Klassen	418
13.11.3	Zweites Teildiagramm: Einbindung der Rangschlüsselklassen	419
13.11.4	Drittes Teildiagramm: Contest-Klassen und deren Nutzer	419
13.12	Implementierung in Java	419
13.13	Operationen und Kontrollen von EvaluateSportsDS (Java)	419
14	File Services	425
14.1	Numerische Datei-Identifikationen	427
14.2	1. Vorschlag	427
14.3	2. Vorschlag	430
15	Test II	437
15.1	The Second-System Effect	437
15.2	Regressionstest	438
15.3	Test der File Services	441
15.3.1	Testleitfaden	442
15.3.2	Test von setupInFiles() mit readFiles = 0	445
15.3.3	Test von setupInFiles() mit readFiles = 2	455
15.3.4	Test von open() für den Sports Dataset	457
15.3.5	Test von open() für den School Sport Results Dataset	463
15.3.6	Fazit	470
15.4	Programmierte Pfadprotokollierung	471
15.4.1	Pfadprotokollierung mit Dateiausgabe	471
15.4.2	Pfadprotokollierung mit relationaler Ausgabe	479
15.4.3	Tracing & Breakpoint Debugging: Risiken und Nebenwirkungen	481
15.4.4	Pfadprotokollierung mit Ausgabe in einen Trace-Speicher	484
16	Test III	495
16.1	The Whole and the Parts	496
16.2	Das Zeiterfassungssystem und EXAMINE	497
16.3	(Integrations-)Test von EvaluateSportsDS	498
17	Ranggruppenverarbeitung III	511
17.1	Einfaches Java-E-Mail-Programm	512
17.2	Datenstrukturen und Ableiten der Programmstruktur	514
17.2.1	Vorläufiges Eingabe-Struktogramm	514
17.2.2	Korrespondenzen zwischen Ein- und Ausgabe	514
17.2.3	Neue Korrespondenztypen: verzögert / erweitert	518
17.2.4	Finale Eingabe-Datenstruktur mit Korrespondenzen	519
17.2.5	Finale Ausgabe-Datenstruktur mit Korrespondenzen	521
17.2.6	Programmstruktur ohne Operationen	525
17.3	Sprachabhängige Aufbereitung	528
17.4	Technisches Format der E-Mail-Daten	531

17.5	Programmstruktur mit Operationen	535
17.5.1	Strukturdefinitionen	535
17.5.2	Definition und Zuordnung von Operationen	538
17.5.3	Top-Struktogramm mit Operationen	541
17.5.4	Ausgelagerte Struktogramme mit Operationen	543
17.6	Durchlaufanalyse	545
17.7	Was hat die Durchlaufanalyse gebracht?	567
17.8	Implementierung in Java	568
18	Anhang	569
18.1	Literaturverzeichnis	569
18.2	Rechtliche Hinweise	571
18.2.1	Geschützte Bezeichnungen	571
18.2.2	Haftungsausschluss	583
18.2.3	Zitate	584
18.2.4	Firmen, Personen und Namen	584
18.2.5	Copyrights	585
18.2.6	Bild- und Grafiknachweis	585
18.2.7	Programme	585