

INHALTSVERZEICHNIS

Vorwort 24

Kapitel 1: Eine Nudelmaschine für zu Hause

Die SAP, das SAP-System und ABAP

Seite 27

Den Schrödinger hat es mal wieder erwischt: Ein österreichischer Nudelhersteller hat seine alte Firma, Spaghetti Infinito, übernommen. Dabei war das doch bis jetzt so gemütlich in der Spaghetti-Informatik. Man konnte programmieren, wie und womit man wollte – Hauptsache, es lief. Wenn er zukünftig keinen Spaghetticode mehr programmieren darf, kann er sich gleich etwas Neues suchen: Zum Beispiel bei der ABAP-Fabrik, von der er schon so viel gehört hat. Klingt lustig, ABAP. Und wenn die auch Code produzieren ..., vielleicht kann er dann ja doch wieder ein bisschen Spaghetticode programmieren! An seinem ersten Tag lernt er den Schwaiger Roland kennen, seinen ABAP-Ausbilder. Der seine Vorliebe für Spaghetticode sofort entdeckt und sich zunutze macht. Aber anders, als Schrödinger denkt.

Servus Schrödinger!!!	28	SAP – SAP-System	38
Schrödingers SAP-System – Eigenes		Die Erfolgsgeschichte: Vom ABAP zum ABAP	41
SAP-System runterladen	31	Kompilieren/Interpretieren	45
Schrödingers SAP-System – Das brauchst du dafür	33	ABAP allgemein	46
Schrödingers SAP-System – Installieren	34	ABAP-Syntax	47
Die SAP – Das Unternehmen	37	Beispiel ABAP	49
		Schrödingers ABAP	50

Kapitel 2: Rein in die Küche – raus aus der Küche

Starten mit dem SAP-System

Seite 51

Ein spannender Tag für den Schrödinger: Das erste Mal! Der Schwaiger Roland meint, das tut gar nicht so weh, wie alle sagen, der hat leicht reden. Und fängt gleich mit der Anmeldung am SAP-System an. Äh, wo eigentlich? Ach ja, SAP Logon. Geschafft, und jetzt das Anmeldebild. Wie? Ach ja, SAP GUI. Aber was soll das mit diesen Mandanten? Wir sind doch keine Anwälte, wir sind Entwickler!

Anmelden zum Tauchkurs: Eintauchen in die SAP-Welt	52	Die Auslage des SAP-Systems – SAP GUI-Aufbau	61
Alles ist Transaktion – Transaktionen als Shortcuts	54	Alles auf einen Blick – SAP Easy Access	63
Melde dich an! – SAP-Anmeldung	55	Modus Operandi – Zusätzliche Fenster öffnen	65
Transaktion starten	57	Hüben und drüben, nichts als Üben – Rolands Folterkammer oder Übung macht den Meister	67
Willkommen Schrödinger! – SAP-Willkommenstext	58		

Kapitel 3: Ciao a tutti! (Hallo Welt!)

Der 20-Minuten-Einstieg in die ABAP-Programmierung

Seite 69

Endlich, das erste ABAP-Programm! Schrödinger war sich bislang gar nicht sicher, ob der Schwaiger Roland überhaupt programmieren kann. Und dann legt er auf einmal richtig los: Pakete anlegen, Datenbankzugriffe, Ausnahmebehandlung, Schlüsselwortdokumentation. Dieses ABAP fängt an, Spaß zu machen, weil es so viel zu entdecken gibt.

Einsteigen und anschnallen!	70	Kühlschrankdesign – DB-Tabelle anlegen	91
Object Navigator – Die integrierte Entwicklungsumgebung	72	Daten verarbeiten – Programm anlegen	95
Entwicklung organisieren – Systemlandschaft, Änderungsauftrag	73	Das ABAP-Einmaleins – Grundlegende Syntax	96
Ihr neuer Auftrag lautet: Auftragsanlage	75	Der Durchblicker – Programm entwickeln	98
Organisationskapsel Paket	77	Her mit den Daten – DB-Zugriff	100
Vom Entwickler zum Pakettier – Paket anlegen	79	Jetzt lese ich	101
Einen Schritt zurück und bald zwei nach vorne	82	Ohne Struktur läuft nichts – Grundlegende Programmstruktur	102
Die Infos zum Aufbau einer Datenbanktabelle ...	83	Alles muss es auch nicht sein – Einfaches Selektionsbild	104
Das zweischichtige Domänenkonzept	84	Layouten und Daten transportieren – Ein einfaches Dynpro	108
Technisches Fundament eines Typs – Domäne anlegen	86	Dekorieren der Auslage – Dynpro-Layout	110
Lege die Bedeutung an – Datenelement anlegen	89	Ablauflogik ohne Ablaufdatum – Ablauflogik programmieren	116
		Ein Shortcut für den User – Transaktionscode anlegen/ausführen	120

Kapitel 4: AAA – Alles außer ABAP

ABAP-Infrastruktur und -Werkzeuge

Seite 121

Der Schrödinger hat es schon gemerkt: Zum ABAPen reicht es nicht aus, sich die Sprache ABAP selbst reinzuziehen. Denn wo liegen die Programme, welche Programme gibt es eigentlich, womit werden sie bearbeitet? Welche Speicherbereiche werden genutzt? Wie kommen die Programme zu den Anwendern? Und ungefähr weitere 1.000 Fragen. Aber der Schwaiger Roland ist ja (noch) geduldig.

Warum?	122	Werkzeugkiste – Entwicklungswerkzeuge	137
Die Ablage der Entwicklungsobjekte –		Object Navigator – Der Regisseur	138
Repository	123	Repository Browser verschwinden lassen	139
Wer sucht der findet – Suchen mit dem		Repository Browser ausnutzen	141
Repository Infosystem	128	Repository Browser browsen	142
Suche von oben – Suchen in der		Synchron oder aus dem Tritt –	
Anwendungshierarchie	130	Objekt-Browser-Synchronisation	145
Suchen und Finden – Repository Infosystem	131	ABAP Editor – Die Schreibmaschine	147
Geschichtet, aber nicht gefaltet –		Debugger – Der Kammerjäger	150
ABAP-Infrastruktur	132	Debugger entschleunigt	151
Wohin mit dem Anwender –		Verwendungsnachweis –	
Speicherorganisation extern und intern	135	Der Rückwärtssucher	155

Kapitel 5: Schräge Typen*

*also jetzt nicht der Schwaiger Roland

Datentypen und -objekte

Seite 157

Definieren und deklarieren: das tägliche Geschäft des ABAP-Programmierers. Das ist ja wie mit den unterschiedlichen Nudeltypen. Die eine Sorte ist lang und dünn, andere sehen aus wie Röhren oder wie Schmetterlinge. Jedoch, und das weiß der Schrödinger natürlich, wird für ein bestimmtes Gericht oder eine besondere Pastasoße ein bestimmter Nudeltyp benötigt. Und so ist es bei der Programmierung eben auch: Für bestimmte Aufgaben werden bestimmte Typen benötigt.

Vorspann	158	Variables Standgas – Datenobjekte	159
Von 0 auf 100 in wenigen Seiten – Technik der		Variablen sind zum Angreifen	160
Datenspeicherung im Hauptspeicher	158	Ein typischer Blickwinkel – Datentypen	161

Anders betrachtet – Datentypen	163	Her mit der internen Tabelle	183
Die Vermessung	164	Wie viel Spalten hätten Sie denn gerne –	
Ansprache – Datenobjekte verwenden	165	Zeilentyp	184
Dynamik pur oder Statik nur? – Statisch und		Normal, sortiert oder doch gehasht –	
dynamisch typisierte Variablen	166	Tabellenart	185
Mein lokaler Typ	167	Open the door please – Schlüssel	187
Zuerst mal elementar lokal – Lokale		Heute mal kopflos – Kopfzeile	188
elementare Typen	168	Tabellen für alle – Globaler Tabellentyp	189
Dann elementar global – Globale		Operation: Table – Tabellenoperationen	191
elementare Typen	170	Bibliothek einräumen	193
Elementar global con domäne – Domäne	173	Einzeln geht es auch – Einzelverarbeitung von	
Strukturell lokal – Lokale strukturierte Typen	176	Tabelleneinträgen	195
Struktur global – Globale strukturierte Typen	178	Ändern und Löschen zum Abrunden –	
Tabellarisch – typisch/intern –		Noch mehr Tabellenoperationen	196
Lokaler Tabellentyp	182		

Kapitel 6: Si parla ABAP? Echtes Küchenlatein

ABAP -Syntax I

Seite 197

Jetzt geht's ans Eingemachte, Schrödinger, oder sollten wir besser sagen ans Eingekochte? Hier lernst du mal das Basisvokabular, so wie damals, als nur Spaghetti gemacht wurden und nicht auch dieses andere Zeugs.

Von Kontrollen und Schleifen	198	Notandor – Logische Ausdrücke	212
Zuweisungen, Operationen, Funktionen	198	Verzweige, falls ... – IF ... THEN ... ELSE	214
Bitte nicht abweisen – Zuweisungen	201	Cassis – CASE ... WHEN ... ENDCASE	216
Was du darfst und was du sollst –		Schleifen	217
Konvertierungsregeln	203	Bedingungslose Schleifen – DO ... ENDDO	217
Ketten ohne Perlen –		Krasses Projekt für Hardcore-ABAPer	218
Zeichenketten-Funktionen	205	Bedingte Schleife – Fußgesteuert	220
Von Segmenten und Offsetten –		Bedingte Schleife – Kopfgesteuert	221
Zeichenketten-Operationen	209	Jetzt aber mal systematisch – Systemfelder	222
Ohne Rechnung, ohne mich – Arithmetische		Großbaustelle Rezeptprogramm	224
Operationen	211		

Kapitel 7: Kleine Häppchen sind leichter wiederverdaut („Schluss mit Spaghetti!“)

ABAP-Syntax II

Seite 227

Schwaiger wirkt wild und entschlossen. Er hat endgültig genug vom Spaghetti-Coding, das Schrödinger fabriziert. „Heute mache ich aus dir einen Wiederverwender“, hallt die letzte Schwaiger-Drohung durch die Hallen, und Schrödinger versucht, sich mit Wehmut von seiner letzten Spaghetti zu trennen ...

Motivation durch Demotivieren	228	Ein Typ muss es sein – Schnittstelle typisieren ...	243
Den richtigen Zeitpunkt finden – Ereignisblöcke	229	Bitte mit Typ – Typisieren	244
Ereignisreiche Programme –		Warum in die Ferne schweifen? –	
Ereignisschlüsselwörter	230	Lokale Variablen	245
Ereignisse: Rekapitulation mit Selbstreflexion	231	Sichtbar oder nicht – Lokale Überdeckung	246
Zur richtigen Zeit an irgendeinem Fleck?	233	Aufruf bitte – Unterprogramm-Aufruf	247
Meine schönsten Ereignisse –		Totalumbau	248
Rahmenprogramm	234	Globale Wiederverwendung –	
Dynpros mit Modul – PBO, PAI, HOV	236	Funktionsbausteine	252
Module, ganz kurz – Dynpro-Programmierung ...	237	Funktionsgruppe	254
Aber nun mit Schnittstelle – Unterprogramme ...	238	Leg dir eine Funktionsgruppe an	257
Definiere zuerst! Unterprogramm-Definition	239	Es ist so weit, global bereit –	
Übergeben und Übernehmen – Schnittstellen-		Funktionsbaustein	258
parameter für das Unterprogramm	240	Her mit dem Rezept	260
Lesen von DB mit Kapsel – DB-Zugriff in		Haaalloooooo – Funktionsbaustein-Aufruf	263
Unterprogramm	242	Dein Finale	265

Kapitel 8: Schrödinger zeigt Klasse

ABAP Objects I

Seite 267

Schrödinger ist heute nicht gut aufgelegt, weil der Schwaiger Roland mit der Objektorientierung daherkommt. Es funktioniert doch alles bestens mit dem „prozeduralen ABAP“ (so nennt das der Schwaiger Roland). Was hat der bloß? Na okay, ein paar Mal haben sich schon Fehler eingeschlichen, die schwer zu finden waren. Mit der Objektorientierung soll ja alles besser werden mit der Ordnung. Und wo mehr Ordnung ist, ist alles leichter zu finden – sogar die Fehler. Und außerdem kann man mit ABAP Objects anscheinend tolle GUIs realisieren. Nach dem ersten Schock-erlebnis ist Schrödinger also gleich überzeugt.

Motivation zur OO	268	Mit Methode – ran an den Code	300
Begriffe aus der Realität	269	Die Schnittstelle der Methode	304
Holistische Sichtweise	271	Ganz spezielle Methoden	307
Modellierung	272	Methoden mit Fülle	309
Erst denken, dann lenken: Einfache UML als		Selbstreferenz	312
Analysewerkzeug zur Visualisierung von		Statisches Attribut	313
Klassen/Objekten	276	Methodenaufruf	314
Klasse Kaffee(-vollautomat) auf dem Papier	278	Laufzeit sequenziert	316
Ohne meine Kollaborateure bin ich nichts	280	Methoden für den Wasserbehälter	318
Klasse oder doch Objekt	282	Von lokal nach global	320
Klasse Lokal	284	Going global – SE24	321
Dein kleines Objektistan	287	Attribute und Methoden	322
Kaffee für zwei	290	Kaffeebohnen im Behälter	326
Datenablage – Attribute	292	Senden und Empfangen	330
Wasserstand und Pause	294	Sender mit Sendungsbedürfnis	331
Klasse Attribute, oder was?	295	Empfänger	332
Privat verkalkt/Private Attribute	298	Objektorientiertes Verkuppeln	333
Ab jetzt mit Methode – Schrödinger frischt auf ...	299	Kaffee s'il vous	335

Kapitel 9: Erben oder nicht, des ist hier die Frage: Pasta di Erbe

Vererbung

Seite 337

Hoffentlich ist der Schrödinger heute in der passenden Stimmung. Eine Vertiefung der Objektorientierung, puh, da wird er schwitzen. „Was, da geht noch mehr? Jetzt hab ich mich gerade erst vom ersten OO-Schock erholt!“ Schrödinger wird wohl erkennen, dass Spaghetti, Ravioli, Tortiglioni etc. alles Nudeln sind, oder?

Motivation	338	Redefinieren – Polymorphie von Methoden	355
Vererben und Erben von Klassen	341	Redefinieren als globale Herausforderung	358
Vererbung lokal	344	Statische und dynamische Typen von Klassen	
Meister der Pyramiden	346	mit ein wenig Casting	364
Globale Vererbung: Ein praktisches Beispiel		Festigung der Einsichten	366
im SAP-Standard	348	Down-Cast	369
Was wird wie und wo vererbt	351	Abstrakte Klassen	372

Kapitel 10: Keine Details bitte

Der Blick durch Interfaces

Seite 377

„Man kann auch noch einen Schritt weiter gehen und die Implementierung von Methoden von deren Schnittstellendefinition trennen“, verkündet Roland und legt eine Kommunikationspause ein. Schrödinger steht da, der Mund ist offen, und er ist sich noch nicht der Tragweite der Aussage bewusst.

Schizophrenie?	378	Von Suchmaschinen und Tilden	387
Lokal zum Einsteigen	381	Hast du gerufen?	388
Schnittig und definiert	382	Die vielen Gestalten der Methoden	390
Volles Risiko – alles oder nichts	383	Globalisierung mit	
Klasse Zuordnung	384	positiven Auswirkungen	393
Klasse Definition	385	Global klasse Implementierung	395
Klasse Implementierung	386	Singleton-Party	397

Kapitel 11: Das Kapitel für Feiglinge

Ausnahmen

Seite 401

„Fehlerbehandlung ist für Feiglinge! Das ist ja so, als ob ich beim Nudelkochen die Nudeln koste, ob sie al dente sind oder nicht!“ Schrödinger hält nichts von Ausnahmebehandlungen. Bis zum ersten Zwischenfall: als seine Freundin sich über klebrige und zerkochte Nudeln beschwerte.

Ausnahmsweise behandelt	402	Ausnahmen melden (mit Nummern)	414
Eine kleine Geschichte der Ausnahmen	404	... und behandeln (nach Nummern)	415
Nochmal langsam: Mit SY-SUBRC.		Ausnahmslos Objekte	417
Ausnahmewert wird gesetzt	408	Definieren geht vor	419
... und von dir behandelt	409	Ausnahmen melden (mit Ausnahmeklassen)	421
It's RAISING man, hallelujah.	410	... und Ausnahmen behandeln, aber richtig!	
Ausnahmen melden	410	Mit Ausnahmeklassen.	424
... und behandeln	411	Das ausnahmengespickte Projekt	425
The Catcher in the System	414		

Kapitel 12: Spaghetti-Programmierung

Selektionsbilder, Listen und Nachrichten (GUI I)

Seite 431

Es ist so einfach, eine SAP-Oberfläche zu programmieren: Ein Selektionsbild erstellen, eine Liste ausgeben, vielleicht noch ein paar Nachrichten an den Anwender verschicken – und fertig ist das berühmt-berüchtigte SAP GUI! Das klingt nach Spaghetticode und ist damit ganz nach dem Geschmack von Schrödinger. Zum Einstieg in die GUI-Programmierung ist es der beste Weg, einfach mal ein paar Elemente auf dem Bildschirm auszugeben. Für das richtige Ambiente sollen dabei Klettererdbeeren und Vogerlsalat sorgen ..., sagt der Schwaiger Roland.

Vom kleinen Window-Farmer bis zum Groß-	Mehr, Mehrere, Mehreres (ohne Rauschen)	453
GUI-Besitzer: Jeder will ein Selektionsbild!	Zur Anwendung gebracht ... fast	456
Selektionsbild für Beschränkte mit Liste	Graue Theorie: Eingabehilfe, Prüftabelle	458
PARAMETERS: Das kleinere der	Entfessele den Künstler in dir:	
Selektionsbild-Ungeheuer	Screen-Gestaltung	460
Typ und Vorschlag	Endlich Texte!	462
Hey, Checker!	Wie im Hamsterrad	464
Rund wie ein Radiobutton	Ereignisreich geht's weiter	470
„Du darfst“ war gestern, heute ist „Du musst“	Tagesschau ... also eine Nachrichtensendung	473
Abflug in den Keller	L – Li – Lis – List – Liste	477
Aber satt war er noch immer nicht:	Keine Beichte notwendig:	
SELECT-OPTIONS	Interaktion mit einer Liste	484

Kapitel 13: Lasagne aufs Auge

Dynpro-Programmierung (GUI II)

Seite 487

„Wow, ein dynamisches Programm!“ Das gefällt Schrödinger sofort, dieses Dynpro. Dynamik ist einfach klasse, außer natürlich, es geht um Sport. Mit der Ablauflogik, Elementen zur Gestaltung der Darstellung und Eigenschaften, die man selbst programmieren kann: Super! Das ist schon richtig modular. (Ja, so langsam gewöhnt sich Schrödinger an die Feinheiten im ABAP-Vokabular.) Und es sind auch schon Schichten erkennbar, die Bausteine zur Dynpro-Programmierung sehen fast schon aus wie eine geschichtete MVC-Architektur. Und Schrödinger liebt Schichten, vor allem in Form von Lasagne!

Dynamisch programmieren	Alles ist im Fluss und manchmal geht es	
Wer schreit hier so? Rahmenprogramme,	im Kreis	497
die Dynpros rufen!	Wohin soll ich mich wenden?	501
Dreieinigkei t	Wo Module bestimmen	502

Mit welchem Titel darf ich dich ansprechen?	505	Dein Dynpro	525
Über der Fläche steht die Oberfläche	508	Sammle Elemente	527
Die Gestaltung eines eigenen Menüs	511	Wohin mit den Daten?	529
Kannst du mit dem Druck umgehen?	514	Ablauflogik Reloaded	531
FFFFF	517	Dynpro rufen mal anders	539
Weck den Künstler in dir	519		

Kapitel 14: Ravioli

Web-Dynpro-ABAP-Programmierung (GUI III)

Seite 543

Heute wartet ein besonderer Leckerbissen auf Schrödinger: Web Dynpro ABAP. Webanwendungen erstellen ohne HTML-Kenntnisse. Das klingt gut, findet Schrödinger. Ist aber gerade nicht so wichtig, findet der Chef. Kann man gut online lesen, finden alle. Also: Schau auf der Bonus-Seite unter <http://www.sap-press.de/3024>

Motivieren funktioniert nicht 544

Bonusseite:
<http://www.sap-press.de/3024>

Kapitel 15: Raus aus meiner Küche!

Berechtigungen

Seite 547

„Hör mal, Schwaiger Roland, zu viele Köche verderben doch bekanntlich die Nudelsoße. Gibt es Möglichkeiten in ABAP, dass nicht jeder alles darf? Also dass ich bestimmten Anwendern manche Aktionen verbieten oder erlauben darf? Wenigstens einschränken?“ Der Schwaiger Roland lächelt nur weise.

Berechtigungsrundumschlag – Überblick	548	Der Berechtigungs-Selbstchecker –	
Berechtigungen		S_TCODE prüfen	556
Am Anfang steht das Objekt mit Klasse –		Experimente mit AUTHORITY-CHECK	558
Berechtigungsobjekt	551	Warum mag mich keiner – SU53	559
Die Details zur Berechtigung	554		

Kapitel 16: Vorratskammer einrichten mit ziemlich viel Schnickschnack

DB-Tabellen erstellen

Seite 561

Was geschieht, wenn Anwender alle Daten im GUI eingegeben haben? Wohin gehen die Daten dann? Die müssen doch gespeichert werden? Denn die Anwender werden wahrscheinlich nicht immer wieder die gleichen Daten eingeben wollen – außer wenn sie an einem schwachen Kurzzeitgedächtnis leiden. Das würde sich Schrödinger öfter wünschen, das mit dem Kurzzeitgedächtnis, denn dann könnten sich die Anwender nicht an seine Programmierfehler erinnern ..., aber der Schwaiger Roland erklärt ihm dann doch lieber, wie er eine SAP-Datenbanktabelle erstellt.

Freiland-Datenhaltung – Daten persistieren	562	Ich will auch anders suchen – Sekundärindex	586
Warum einfach, wenn es mit Schnittstelle geht –		Ändern oder nicht, was geht –	
Die Datenbankschnittstelle	564	Erweiterungskategorie	589
Transparente Tabelle en Detail	567	Definieren und Realisieren –	
Spalten und der Rest – Tabellenfelder	570	Datenbankobjekt	591
Ohne Technik keine Tabelle – Technische		Mein erster Eintrag – Datenbanktabellen-	
Einstellungen	575	Einträge erzeugen	592
Mehr als eine Tabelle	578	Artenvielfalt im Dictionary –	
Welcher Wert ist möglich – Fremdschlüssel	581	Weitere Tabellenarten	594

Kapitel 17: Vorratskammer plündern

DB-Tabellen auslesen

Seite 595

Daten rein, Daten raus, und das möglichst einfach: So wünscht sich das der Schrödinger. Da kann ihm der Schwaiger Roland helfen. Und alleine die Begriffe „Open SQL“ und „ANSI SQL“ klingen wie Musik in seinen Ohren. Und er kann dann Daten aus einer oder sogar mehreren Datenbanktabellen lesen. Manchmal wünscht sich Schrödinger, er könnte den Schwaiger Roland in der Vorratskammer einsperren. Nur über Nacht.

Erster Takt – SQL	596	Alles recht und schön – Berechtigungen und	
Zweiter Takt – SQL	596	Konsistenzprüfungen	598
Eingelagert und geplündert – Datenpufferung	597	Open SQL grundiert – Basisbefehlssatz aus	
Datenmanipulator nativ – DML mit Native SQL ...	598	Open SQL	599
Einfach definieren – Datendefinition mit		Verklaussliert – SELECT-Klausel	601
dem ABAP Dictionary	598	Wohin damit – INTO target-Klausel	604

Woher – FROM source-Klausel	605	Da will ich auch nicht alles –	
Aber bitte nicht alles – WHERE-Klausel	607	Selektionsbedingungen	627
Da geht noch mehr – Weitere Klauseln	608	Zusammenfassung in Bildern –	
Und jetzt alles kombiniert – SELECT-Beispiel	609	View-Definition	628
Mehr als eine Tabelle ist auch okay – Views	614	Mit Views programmieren	630
Schritt für Schritt zum View mit relationalen		Weil du so brav warst – Programmieren	
Operatoren	618	mit Views	632

Kapitel 18: Vorratskammer in Schuss halten

DB-Daten pflegen

Seite 635

Daten pflegen – den Ausdruck mag Schrödinger. Und denkt an die Kräutersammlung auf seinem Fensterbrett. Da müssen manchmal auch neue Kräuter eingefügt werden, mal muss er das eine durch das andere austauschen. Und für das Löschen hat er auch schon eine leckere Soßenidee. Datenpflegeservice Schrödinger!

Lesen ist Silber, Schreiben ist Gold –		Und noch eins und noch eins und ... – INSERT ...	650
DELETE, UPDATE, MODIFY und INSERT	636	Massendaten einfügen	652
Sichere Daten – Transaktionskonzept	636	Ändern muss man auch ab und zu – UPDATE	655
Von einem Zustand zum nächsten –		Massendaten	656
Datenbank-LUW	638	Weg damit – DELETE	658
Bist du konsistent?	641	Kombianweisung – MODIFY	661
Aktionen bündeln – SAP-LUW	642	Halt, Sperre! – Sperrkonzept	663
Und jetzt auch noch transaktional –		Die fünf Gebote der performanten	
SAP-Transaktion	643	DB-Programmierung	664
Datenmanipulator-Entwicklungsrahmen –			
Das Programm für die Open-SQL-Anweisungen ...	645		

Kapitel 19: Mit Schirm, Charme – und vielleicht noch einem Melonensorbet

Daten in Dateien, Datenablage ohne DB

Seite 665

„Warst du schon einmal bei einem richtig guten Italiener?“ Schrödinger ahnt nichts Gutes. „Also so einem, wo du bereits am Eingang dem Kellner deine Garderobe abgeben kannst.“ – „Ja, das Prinzip der Garderobe ist mir bekannt, wieso?“ – „Na ja, wäre es nicht klasse, wenn die Anwender in ABAP auch Dateien einlesen oder herun-

terladen könnten – also an der Garderobe abgeben könnten?" Au backe, das ist ja mal eine miese Metapher. Aber es ist vielleicht trotzdem wichtig zu wissen, wie man mit Daten umgeht, wenn man kein Datenbanksystem hat.

Daten ohne Datenbank	666	Download now!	676
GUI-Loads – Upload und Download	666	Einen hamma noch – Upload	679
Frontend Services	668	Daten auf dem Applikationsserver	681
Pfad ermitteln	671	OPEN House – OPEN DATASET	682
Download now, zumindest vorbereitet –		DatenTRANSFER – Daten schreiben	683
Download vorbereiten	673	READ DATASET – Daten lesen	685
Auch nett für Datenwiederverwendung –			
Clipboard	674		

Kapitel 20: Täglich wechselnde Speisekarten

Dynamische Programmierung

Seite 687

Morgens frische Lebensmittel auf dem Markt einkaufen, mittags daraus eine Speisekarte zaubern. Jeden Tag neu. Schrödinger bekommt Hunger, aber darum kann es dem Schwaiger Roland jetzt ja nicht gehen. „Was genau ...“ – „Wenn du erst während der Laufzeit eines Programms die Informationen erhältst, die für die Ausführung des Programms nötig sind, zum Beispiel den Namen einer Datenbanktabelle, dann ...“ – „Okay, okay, ein neues Konzept. Sag mir, wie es heißt, aber dann geht es in die Kantine.“

Dynamische Programmierung hat nichts mit		Dynamisches Feld	697
Beweglichkeit zu tun	688	Dynamischer Typ	697
Feldsymbole	688	Dynamische Komponente	698
Datenrefs	690	Dynamische Bedingung	698
RTTS = RTTI + RTTC	692	Dynamisches Unterprogramm	700
Überschrift finden	693	Programmerzeugung im Hauptspeicher	700
Dynamische Tokens	695	Programmerzeugung im Repository	701

Bildnachweis	704
--------------------	-----

Index	705
-------------	-----