

Ulrich Mende

Moderne Workflow- Programmierung mit ABAP® Objects

Handbuch für Entwickler



dpunkt.verlag

Ulrich Mende
ulrich.mende@web.de

Lektorat: Dr. Michael Barabas
Copy-Editing: Ursula Zimpfer, Herrenberg
Satz: Josef Hegele, Birgit Bäuerlein
Herstellung: Birgit Bäuerlein
Umschlaggestaltung: Helmut Kraus, www.exclam.de
Druck und Bindung: M.P. Media-Print Informationstechnologie GmbH, 33100 Paderborn

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

ISBN 978-3-86490-013-6

1. Auflage 2012
Copyright © 2012 dpunkt.verlag GmbH
Ringstraße 19 B
69115 Heidelberg

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

In dieser Publikation wird auf Produkte der SAP AG, Dietmar-Hopp-Allee 16, 69190 Walldorf/Deutschland, Bezug genommen. Bei den Bezeichnungen dieser Produkte handelt es sich um eingetragene und/oder nicht eingetragene Marken der SAP AG. Die SAP AG ist weder Autor noch Verleger dieses Buches und ist für seinen Inhalt nicht verantwortlich.

Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autor noch Verlag können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.

5 4 3 2 1 0

Inhaltsverzeichnis

	Einleitung	1
	Wer hat das Buch geschrieben?	1
	Dank	1
	Worum geht es in diesem Buch?	2
	An wen richtet sich das Buch?	3
n	Hinweise zu den einzelnen Kapiteln	4
1	SAP Business Workflow im Überblick	11
1.1	Die Architektur des SAP Business Workflow	13
1.1.1	Definitionszeit	13
1.1.2	Workflows und Workitems zur Ausführungszeit	16
1.1.3	Bearbeiterfindung	16
1.1.4	Workflow als Reaktion auf Anwendungssignale	17
1.1.5	Vorkonfigurierte Workflow-Definitionen	17
1.1.6	Der typische Workflow-Benutzer	18
1.2	Ein typisches Workflow-Szenario	19
1.3	Dokumentation und Tutorial	20
1.4	Business Workplace	20
1.4.1	Mails im Business Workplace	21
1.4.2	Workitems mit Benutzerdialog in der Worklist	21
2	Workflow-Entwicklung – Tools	23
2.1	Workflow aus Entwicklersicht	23
2.1.1	Workflow als Framework	23
2.1.2	Schichtenmodell einer Workflow-Anwendung	23
2.1.3	Paketorientierte Workflow-Entwicklung in der SE80 ...	24
2.1.4	Das Workflow-Bereichsmenü SWLD	25

2.2	Workflow-Beispiellösungen der SAP	26
2.2.1	Tutorials	26
2.2.2	Workflow-Anwendung »Urlaubsantrag«	27
2.2.3	Workflows der Aufgabengruppe WFUNIT	27
2.2.4	Workflows der Aufgabengruppe WF_Verify	28
2.3	SAP Service Marketplace	29
2.3.1	SAP Support Portal mit Hinweissystem	29
2.3.2	SDN-Community	32
2.3.3	SAP Help online und offline	33
3	ABAP Objects – Grundlagen	35
3.1	Objektorientierung als Abbild der realen Welt	35
3.2	Klassen und ihre Komponenten	36
3.2.1	PUBLIC-, PROTECTED- und PRIVATE-Komponenten	36
3.2.2	Statische und Instanzkomponenten	37
3.2.3	Abstrakte und finale Methoden und Klassen	37
3.2.4	Lokale und globale Klassen	38
3.2.5	Objekte und Referenzen	40
3.2.6	Zugriff auf Klassen- und Instanzkomponenten	42
3.3	Vererbung und Casting von Referenzen	44
3.3.1	Klasse CL_VATER	46
3.3.2	Klasse CL_SOHN	48
3.3.3	Klasse CL_ENKEL	49
3.3.4	Beispielprogramm zur Vererbung	50
3.3.5	Upcasting und Downcasting von Referenzen	52
3.4	Events	55
3.4.1	Prinzip Publish and Subscribe	55
3.4.2	Event Handling	56
3.4.3	Events im Workflow-Kontext	57
3.5	Interfaces	57
3.5.1	Definition von Interfaces	59
3.5.2	Interfaceimplementierung und -referenzen	59
3.6	Polymorphie	61
3.6.1	Allgemeines	61
3.6.2	Polymorphie durch Vererbung	63
3.6.3	Polymorphie über Interfaces	64
3.7	Ausnahmen und Ausnahmeklassen	64
3.7.1	Klassische Ausnahmen ohne Ausnahmeklassen	64
3.7.2	Ausnahmeklassen	66
3.7.3	Ausnahmetexte aus dem OTR	75

3.8	Zugriff auf die Klassendefinition im Repository	76
3.9	Dynamischer Aufruf von Klassenmethoden	78
3.9.1	Parameter vom Typ REF TO dynamisch übergeben	82
3.10	Run Time Type Services	86
3.10.1	Run Time Type Identification (RTTI)	86
3.10.2	Run Time Type Creation (RTTC)	89
3.11	Übernehmen lokaler Klassen in das Repository	91
4	Durchgängiges Beispiel des Buches: Kreditverträge	93
4.1	Aufgabenstellung	93
4.2	Tabelle ZVERTRAG	93
4.3	Vertragstransaktion	94
4.4	Statusdiagramm	94
4.5	Implementierung (DB und ABAP)	95
4.6	Quelltext der Vertragstransaktion	98
4.7	Quelltext zur Berechnung des Ratenplans	102
5	Die neue Welt: ABAP Objects im Workflow	107
5.1	Problemstellung	107
5.2	IF_WORKFLOW	108
5.3	Vertragsbearbeitung im Workflow – Klasse ZCL_VERTRAG ...	112
5.3.1	Attribute MV_VERNR und MS_VERTRAG	112
5.3.2	Methoden ANZEIGEN, AENDERN, GENEHMIGEN .	113
5.3.3	Ereignisse CREATED, CHANGED und CANCELLED .	115
5.3.4	Erste Tests der Klasse ohne Interface IF_WORKFLOW	116
5.4	Einfachste Ausprägung von IF_WORKFLOW	116
5.5	Erweiterte Ausprägung IF_WORKFLOW	119
6	Workflow-Container und Workflow-Ereignisse	125
6.1	Konzept	125
6.2	Workflow-Container	127
6.2.1	Die Klasse CL_SWF_CNT_CONTAINER	127
6.2.2	Datenflüsse zwischen Containern	133
6.2.3	Containerpersistenz auf der Datenbank	136
6.3	Ereigniserzeugung	141
6.3.1	Universelle Ereigniserzeugung in ABAP-Programmen ..	141
6.3.2	Erzeugung von BOR-Events mit SWE_EVENT_CREATE	145
6.3.3	Ereigniserzeugung über Änderungsbelege	145
6.3.4	Weitere Methoden der Ereigniserzeugung	150

6.4	Ereignis-Verbraucher-Kopplungen	150
6.4.1	Typkopplung mit Funktionsbaustein oder Methode	151
6.4.2	Verbrauchertyp	153
6.4.3	Verbraucher-Funktionsbaustein	153
6.4.4	Check-Funktionsbaustein	153
6.4.5	Verbrauchertyp-Funktionsbaustein	154
6.4.6	Typkopplung aktiv, Ereignis-Queue zulassen	155
6.4.7	Instanzkopplung	155
6.4.8	Einträge in den Koppeltabellen durch Workflow	156
6.4.9	Workflow-Startbedingungen	157
6.5	Ereignisverarbeitung	159
6.5.1	Ablauf der Ereignisverarbeitung	159
6.5.2	Erzeuger- und Verbraucherkontext	159
6.5.3	Ereignisfluss bei Vererbung	160
6.6	Laufzeitwerkzeuge für Ereignisse	161
6.6.1	Ereignis auslösen	161
6.6.2	Ereignis simulieren	162
6.6.3	Ereignis-Trace	163
6.6.4	Ereignis-Queue	164
7	Organisationsmanagement und Regeln	167
7.1	Konzept	167
7.2	Bestandteile des Organisationsmanagements	168
7.2.1	Organisationseinheiten	168
7.2.2	Besetzungsplan	168
7.2.3	Aufgabenprofil	169
7.3	Einfache Pflege der Aufbauorganisation	170
7.4	Infotypen	171
7.4.1	OM-Typeebene und OM-Objektebene	171
7.4.2	OM-Objekttypen (Infotyp 1000)	173
7.4.3	Verknüpfungen zwischen Objekttypen (Infotyp 1001) . .	173
7.4.4	Weitere Infotypen	175
7.4.5	Zuständigkeiten	175
7.5	Regeln und Regelauflösung	176
7.5.1	Konzept	176
7.5.2	Regeldefinition	177
7.5.3	Regeln mit Funktionsbausteinen	178
7.5.4	Regeln mit Zuständigkeiten	179

7.6	Dynamische Eingangskörbe	182
7.6.1	Aufgabenstellung	182
7.6.2	Das Prinzip der dynamischen Eingangskörbe	183
7.6.3	Pflege der Bestandteile dynamischer Eingangskörbe ...	185
8	Standardaufgaben anlegen und verwenden	189
8.1	Konzept Standardaufgaben	189
8.2	Standardaufgaben anlegen	190
8.2.1	Objekttyp und Methode	190
8.2.2	Hintergrundaufgabe und Dialogaufgabe	192
8.2.3	Verarbeitungsende bestätigen	193
8.2.4	Container und Initialwerte	194
8.2.5	Aufgabe mit Rückgabewert	196
8.2.6	Workitem-Text	199
8.2.7	Beschreibung	200
8.2.8	Auslösende Ereignisse	201
8.2.9	Beendende Ereignisse	202
8.2.10	Mögliche Bearbeiter	203
8.2.11	Zuständige Bearbeiter – Defaultregeln	205
8.3	Standardaufgaben starten	206
8.3.1	Start aus Workflow (Standardfall)	206
8.3.2	Start über Ereignis	206
8.3.3	Start aus Programm	206
9	Workflow-Muster anlegen und verwenden	209
9.1	Konzept Workflow-Muster	209
9.2	Workflow-Interface	210
9.2.1	Workflow-Container	211
9.2.2	Initialwerte	213
9.2.3	Aufgabentext	213
9.2.4	Aufgabenbeschreibung	213
9.2.5	Auslösende Ereignisse	214
9.3	Workflow-Definition	215
9.3.1	Die Arbeit mit dem grafischen Editor	215
9.3.2	Der Bedingungseditor	217
9.3.3	Übersicht Schrittypen	218
9.3.4	Schrittyp Aktivität	220
9.3.5	Schrittyp Webaktivität	223
9.3.6	Schrittyp Mailversand	223
9.3.7	Schrittyp Formular	224
9.3.8	Schrittyp Benutzerentscheidung	225

9.3.9	Schritttyp Warten auf Ereignis	226
9.3.10	Schritttyp Bedingung	227
9.3.11	Schritttyp Mehrfachbedingung	228
9.3.12	Schritttyp Paralleler Abschnitt	229
9.3.13	Schritttyp UNTIL-Schleife	230
9.3.14	Schritttyp WHILE-Schleife	232
9.3.15	Schritttyp Ablaufsteuerung	233
9.3.16	Schritttyp Containeroperation	234
9.3.17	Schritttyp Ereigniserzeuger	235
9.3.18	Schritttyp Ad-hoc-Anker	235
9.3.19	Schritttyp Block	235
9.3.20	Schritttyp Lokaler Workflow	236
9.4	Container in der Workflow-Definition	236
9.5	Workflow-Grunddaten	237
9.5.1	Versionsunabhängig (Aufgabe)	238
9.5.2	Versionsabhängig (aktuelle Workflow-Version)	240
9.6	Einfacher Vertrags-Workflow	243
9.6.1	Technische Workflow-Beschreibung und Pseudocode	243
9.6.2	Workflow-Muster zao_main	244
9.7	Bearbeiterfindung im Workflow	249
9.7.1	Allgemeines	249
9.7.2	Definition der möglichen Bearbeiter	249
9.7.3	Zuständige und ausgeschlossene Bearbeiter	250
9.7.4	Ermittlung der Empfänger	251
9.8	Arbeiten mit dem Business Workflow Explorer	252
9.8.1	Zweck des Business Workflow Explorer	252
9.8.2	Auswahl des Suchbereiches	253
9.8.3	Anzeige und Bearbeitung von Objekten	254
9.9	Lesen der Workflow-Definitionen	254
9.10	Starten von Workflows	257
9.10.1	Start von Workflows über ABAP OO	257
9.10.2	Start von Workflows über Ereignisse	258
10	Klassenbasierte Techniken im Workflow	259
10.1	SAP-Klassen im Workflow-Umfeld	259
10.2	Iterator-Klassen, Factory-Klassen und Utility-Klassen	260
10.2.1	Iterator-Klassen	260
10.2.2	Factory-Klassen und Factory-Methoden	263
10.2.3	Utility-Klassen	265

10.3	Terminüberwachung	266
10.3.1	Problem und Lösungsprinzip	266
10.3.2	Standardfall mit Deadline-Workitem	266
10.3.3	Berechnete Termine mit Deadline-Workitem	268
10.3.4	Obsolet setzen eines Workitems	269
10.4	Nebenmethoden	270
10.4.1	Problem und Lösungsprinzip	270
10.4.2	Methode zur Vertragsanzeige	271
10.4.3	Eintrag als Nebenmethode in die Aktivität	271
10.4.4	Amodaler Aufruf der Nebenmethode	272
10.5	Objektreferenzen in die Workitem-Vorschau einstellen	273
10.5.1	Referenzen auf bestehende Workflow-Klassen	273
10.5.2	Referenzen auf spezielle Anzeigeklassen	274
10.6	Programmier-Exits für Workitems	279
10.6.1	Das Interface IF_SWF_IFS_WORKITEM_EXIT	279
10.6.2	Logging mit CHECK-POINT-Gruppe	279
10.6.3	Automatisches Zurücklegen eines Workitems	282
10.7	Fortgeschrittene Workflow-Definition	284
10.7.1	Lokale Ereignisse und Ausnahmen	284
10.7.2	Lokale Ereignisse und komplexe Bedingungen	285
10.7.3	TRY/CATCH im Workflow: Blöcke und Ausnahmen	287
10.7.4	Lokale Workflows und lokale Ereignisse	287
10.8	Dynamische Blöcke	288
10.8.1	Dynamisch sequenzieller Block = Workflow-Iterator	289
10.8.2	Dynamisch paralleler Block	289
10.8.3	Beispiel: Bewertungs-Workflow für Vertrag	290
10.9	Ad-hoc-Funktionen im Workflow	294
10.9.1	Ad-hoc-Aufgabenauswahl	294
10.9.2	Ad-hoc-Bearbeiterauswahl	296
10.9.3	Ad-hoc-Workflow einbinden	298
10.10	Reaktion eines Workflows auf externe Ereignisse	300
10.10.1	Standardreaktionen über Grunddaten des Workflows	301
10.10.2	Modellierte Reaktionen über Ereignisempfänger	302
10.11	Klassenverwendung im Workflow ohne Standardaufgaben	304
10.11.1	Konzept	304
10.11.2	%-%-Zugriff ohne IF_WORKFLOW	305
10.11.3	&&-Zugriff über ungebundene Containerreferenzen	306
10.11.4	&&-Zugriff über gebundene Containerreferenzen	307
10.11.5	Geschachtelter Zugriff mit %-% und &&	309
10.11.6	Methoden mit mehreren Exportparametern	310
10.11.7	Objektreferenzen on-the-Flight	311

11	Klassenbasierte Eigenentwicklungen im Workflow	313
11.1	Utility-Klasse ZCL_DATE: Datumsberechnung auf Kalendern ...	313
11.1.1	Verwendung von Datumsberechnungen	313
11.1.2	Implementierung von ZCL_DATE	313
11.1.3	Verwendung von ZCL_DATE in Ausdrücken	316
11.2	Utility-Klasse ZCL_COND: komplexe Bedingungen	317
11.2.1	Konzept komplexer Bedingungen im Workflow	317
11.2.2	Tabelle ZCONDITION	318
11.2.3	Implementierung von ZCL_COND	318
11.2.4	Verwendung von ZCL_COND in ZCL_VERTRAG ...	319
11.2.5	Setzen und Prüfen von Vertragsbedingungen	320
11.2.6	Aufruf von ZCL_COND in Ausdrücken	325
11.3	Utility-Klasse ZCL_CONST: Datenbankkonstanten	325
11.3.1	Konzept »variabler« Konstanten	325
11.3.2	Tabelle der Konstanten ZCONST	326
11.3.3	Aufruf der Konstanten über ZCL_CONST	326
11.3.4	Implementierung von ZCL_CONST	327
11.3.5	Einführen neuer Konstanten	329
11.3.6	Verwendung der Konstanten im Workflow	329
11.4	Utility-Klasse ZCL_SYSTEM: generische Systemmethoden	331
11.4.1	Inhalt von ZCL_SYSTEM	331
11.4.2	Erzeugen von Klassenreferenzen im Workflow	332
11.4.3	GENERICINSTANTIATE von CL- und BOR-Referenzen	334
11.4.4	MAKE_CL_REF für den Aufruf in Ausdrücken	335
11.5	Kommunikation zwischen parallelen Subworkflows	336
11.5.1	Kommunikation über Ereignisse	337
11.5.2	ZCL_SYSTEM: Lesen fremder Workflow-Container ...	338
11.6	Mails im Workflow-Umfeld	341
11.6.1	Mail senden mit Schritttyp Mailversand	341
11.6.2	ZCL_SYSTEM: Mail mit eigenem Absender	342
11.6.3	ZCL_SYSTEM: Mail mit SAP – Text und Containerbezug	344
11.6.4	HTML-Mails mit Rückantwort	349
12	Business Workplace als Userinterface	355
12.1	Allgemeines zum Business Workplace	355
12.2	Fenster im Business Workplace	357
12.2.1	Das Mappenfenster (1)	357
12.2.2	Das Inboxfenster (2)	360
12.2.3	Das Vorschaufenster (3)	361

12.3	Funktionen im Workplace	362
12.3.1	Funktionen der Workflow-Inbox	362
12.3.2	Persönliche Workflow-Einstellungen	367
12.3.3	Workflow-Vertretungen	367
12.3.4	Vertretungen in Tabelle HRUS_D2	370
12.4	Die Anpassung der Inbox	370
12.4.1	Layoutfunktionen des ALV-Grids	370
12.4.2	Verwendung dynamischer Anwenderspalten	373
13	Webservices im Workflow	377
13.1	Webservices im Workflow	377
13.2	Das Webservice-Framework	378
13.2.1	Erstellen eigener Webservices	379
13.2.2	Einbinden fremder Webservices	380
13.3	Beispiel: Einbinden eines Webservice	381
13.3.1	SOAP-Webservice für Wetterdienste	381
13.3.2	Einbinden eines Webservice in SAP	382
13.3.3	Generierte Klasse ZCO_GLOBAL_WEATHER_SOAP ..	385
13.3.4	Anlegen logischer Ports	387
13.4	Aufruf des Webservice	389
13.4.1	Aufruf des Service GlobalWeather in einem Programm ..	389
13.4.2	Einbinden des Webservice in den Workflow	391
14	Das Workflow-Laufzeitsystem	393
14.1	Konzept	393
14.2	Die Tabellen des Laufzeitsystems	393
14.2.1	Die Tabelle SWWWIHEAD der Workitem-Köpfe	394
14.2.2	Die Tabelle SWW_WI2OBJ für Objektreferenzen	396
14.2.3	Die Tabelle SWWORTASK	397
14.2.4	Die Tabelle SWWUSERWI der Benutzerzuordnung ...	397
14.2.5	Views	397
14.2.6	Programmierter Zugriff auf die Laufzeittabellen	397
14.3	Wichtige Funktionsbausteine und Klassen	400
14.3.1	SAP_WAPI-Bausteine	400
14.3.2	Klassen des Laufzeitsystems	401
14.4	Die Arbeitsweise des Laufzeitsystems	402
14.4.1	Datenfluss über Container	406
14.4.2	Aktionen auf Workitems	406
14.4.3	Termin- und Fehlerüberwachung	407
14.4.4	Business Workplace als Sicht auf die Laufzeittabellen ..	407

15	Tests und Fehlersuche	409
15.1	Workflow-Instanzen suchen	409
15.1.1	Workitem-Selektion mit ihren Anzeigemöglichkeiten . . .	409
15.1.2	Workflow-Instanzen zu Objekttyp	411
15.1.3	Workflow-Instanzen zu Objektinstanz	412
15.2	Workflow-Protokoll mit seinen diversen Sichten	413
15.2.1	ActiveX-Protokoll und seine Sichten	414
15.2.2	Technisches Workflow-Protokoll und seine Sichten	416
15.2.3	Grafisches Workflow-Protokoll	417
15.2.4	Anzeigen, Ändern und Beenden von Workitems	417
15.2.5	Testen der Fristverletzung an Ereignisempfängern	418
15.3	Debug-Report für Batch-Workitems	420
15.3.1	Notwendigkeit	420
15.3.2	Anwendung	421
15.3.3	Quelltext des Reports ZAO_DEBUGWI	422
15.4	Hintergrundschrte im Status STARTED	424
15.4.1	Dump-Analyse	424
15.4.2	Wiederanstarten nach Fehlerkorrektur	426
15.5	Nützliche Hilfsprogramme	426
15.5.1	Container-Update	426
15.5.2	SBE fortsetzen mit Simulation der Auswahl	427
15.5.3	Benutzer an Dialog-Workitems ändern	428
15.5.4	Ultima Ratio nach 22 Uhr	430
15.6	Up- und Download von Workflow-Mustern	434
15.6.1	Problematik	434
15.6.2	Noch kein SAPlink-Plug-in für Workflows	435
15.6.3	Up- und Download von Klassen im XML-Format	436
15.6.4	Up- und Download von Workflow-Mustern als XML . .	436
15.7	Workflow-Transporte	438
16	Workflow-Administration	439
16.1	Aufgabe und Festlegung der Administratoren	439
16.1.1	Allgemeine Aufgaben des Workflow-Administrators . . .	439
16.1.2	Customizing verifizieren	440
16.1.3	Workflow-Administrator global einstellen	441
16.1.4	Workflow-Administrator pro Workflow-Definition	441
16.2	Batchjobs konfigurieren und einplanen	442
16.2.1	Terminüberwachung	442
16.2.2	Workitem-Fehlerüberwachung	443
16.2.3	Bedingungsauswertung	443

16.3	E-Mail-Notification für SAP-Gelegenheitsbenutzer	444
16.3.1	Pflege der E-Mail-Adresse für das Weiterleiten	444
16.3.2	Pflege der persönlichen E-Mail-Adresse in den Benutzerdaten	444
16.3.3	Erzeugen von E-Mails zu neuen Workitemen	445
16.3.4	Versenden der Internetmails über SAPconnect	447
16.4	Reaktion auf Fehlermeldungen des Laufzeitsystems	448
16.5	Typische Workflow-Fehler und ihre Behebung	449
16.5.1	Workitemen ohne Bearbeiter	449
16.5.2	Workitemen mit Terminüberschreitungen	450
16.5.3	Diagnose fehlerhafter Workitemen	450
16.5.4	Laufzeitpuffer synchronisieren	451
16.6	Restarten unterbrochener Workflows	452
16.6.1	Restart fehlerhafter Workflows	452
16.6.2	Restart nach Systemabsturz oder Schrittdump	452
16.7	Analysen	453
16.7.1	Workitemen pro Aufgabe	453
16.7.2	Workitemen nach Bearbeitungsdauer	454
16.7.3	Workload-Analyse	455
16.8	Reorganisation der Laufzeitdaten	457
16.8.1	Allgemeines	457
16.8.2	Auslastung der Workflow-Laufzeittabellen	457
16.8.3	Gezieltes Löschen von einzelnen Workflows	458
16.8.4	Löschen von Workflows im Batch	458
16.8.5	Archivieren von Workflows im Batch	462
16.9	Transportieren von Workflows	465

Anhang

A	Die alte Welt – BOR-Objekte im Workflow	467
A.1	Historisches	467
A.2	Anlegen eines neuen Objekttyps	468
A.2.1	Die Objektliste mit ihren Bestandteilen	469
A.2.2	Programm des Objekttyps als Subroutinenpool	471
A.2.3	Interface	472
A.2.4	Freigeben von Objekttyp und Komponenten	473
A.3	Attribute	474
A.3.1	Datenbankattribute mit GET_TABLE_PROPERTY	474
A.3.2	Objektreferenzen und Makros für den Objektzugriff	477
A.3.3	Implementierung und Aufruf virtueller Attribute	479

A.4	Implementierung und Aufruf von Methoden	484
A.4.1	Synchrone Anzeigemethode als Defaultmethode	487
A.4.2	Synchrone Dialogmethode mit Ergebnisparameter	488
A.4.3	Synchrone Hintergrundmethode mit Parametern	491
A.4.4	Asynchrone Dialogmethode ohne alles	492
A.4.5	Methoden mit Ausnahmen	493
A.4.6	Überdefinieren von Methoden	495
A.5	Ereignisse	495
A.6	Prüfen und Testen einer Objektimplementierung	496
A.7	Vererbung und Delegation	498
A.7.1	Vererbung und Delegation im BOR	498
A.7.2	Vererbung und keine Delegation für ABAP-Klassen	501
A.7.3	Ereignisbehandlung bei Vererbung und Delegation	501
A.8	Standardisierte Schnittstellen – Container	502
A.8.1	Container in Programmen – Include <CNTN01>	503
A.8.2	Persistente Instanzen von Containern	507
A.8.3	Container im Workflow	508
A.9	Spezielle Objekttypen	509
	Literatur	511
	Inhalt des Downloadbereiches	513
	Abkürzungsverzeichnis	517
	Index	519