

Inhaltsverzeichnis

Einführung	15
1 Workbench-Werkzeuge und Pakethierarchie	25
1.1 Anmelden und erste Schritte	26
1.1.1 Workbench Object Browser	26
1.1.2 Object-Browser-Liste	27
1.1.3 Workbench-Einstellungen	30
1.1.4 Neuer ABAP Editor	31
1.1.5 Weitere neue Workbench-Werkzeuge	35
1.1.6 Debugger	38
1.2 Pakethierarchie für das Projekt	42
1.2.1 Projektanforderungen	43
1.2.2 Pakethierarchie	44
2 Data-Dictionary-Objekte	47
2.1 Entwerfen von Datenbeziehungen	47
2.1.1 Grafische Darstellung der Tabellenbeziehungen	49
2.1.2 SAP Data Modeler	50
2.2 Data-Dictionary-Domänen	51
2.2.1 Einzelwerte	52
2.2.2 Festwertintervalle	53
2.3 Data-Dictionary-Texttabellen	54
2.3.1 Datenelemente und Domänen	54
2.3.2 Erstellen von transparenten Tabellen und Beziehungen	55
2.3.3 Pflege-View	61
2.3.4 Generierte Tabellenpflege	62
2.4 Data-Dictionary-Datentabellen	65
2.4.1 Erweiterungen	65
2.4.2 Indizes	66
2.4.3 Sperren von Objekten	69
2.5 Suchhilfen	71
3 Datenpersistenz-Schicht	75
3.1 Persistente Objekte	76
3.1.1 Erstellen der persistenten Objektklasse	77

3.1.2	Basisklassen und Agentenklassen	79
3.1.3	Persistent Data Mapper	81
3.1.4	Coding für persistente Objekte	84
3.2	Ausnahmeklassen	87
3.2.1	Vorteile von Ausnahmeklassen	87
3.2.2	Erstellen einer Ausnahmeklasse	89
3.3	Business-Objekt-Klassen	94
3.3.1	Was ist eine Business-Objekt-Klasse?	94
3.3.2	Business-Objekt-Klassenstruktur	97
3.3.3	Auswahl mehrerer Objekte	100
3.3.4	SELECT-OPTIONS als Query-Kriterium	102
3.3.5	Komplexe Business-Objekte	106
3.3.6	Änderungsoperationen	109
3.3.7	ZIP-Komprimierung	110
3.4	Datenübernahmeprogramme	113
3.4.1	Generieren von Testdaten	113
3.4.2	Sicherungs- und Wiederherstellungsprogramm	118

4 Verwenden eines Webservice 125

4.1	Proxy-Generierung	127
4.1.1	Zugreifen auf das WSDL-Dokument	128
4.1.2	Erstellen des Clientproxys	128
4.1.3	Logische Ports	132
4.1.4	Verwenden des Clientproxy-Objekts	133
4.1.5	Implementierung in der Business-Objekt-Schicht	134
4.2	Logische Ports	137
4.2.1	Laufzeit	137
4.2.2	Aufrufparameter	139
4.2.3	Operationen	143
4.2.4	Fehleranalyse	143
4.2.5	PI-Empfänger	144
4.2.6	Applikationsspezifische globale Einstellungen	145
4.2.7	Applikationsspezifische Operationen	146
4.3	Erweiterte Protokolle	147

5 Shared Memory Objects 149

5.1	Erste Schritte	150
5.1.1	Erstellen der Gebietswurzelklasse	150
5.1.2	Definieren des Shared-Memory-Gebiets	157

5.1.3	Testen des Shared Memory Objects	158
5.1.4	Shared Memory Monitor	160
5.2	Automatisches Vorabladen	161
5.2.1	Hinzufügen der Schnittstelle	162
5.2.2	Modifizieren des Leseprogramms	164
5.3	Implementierung in der Business-Objekt-Schicht	164
5.3.1	Entwickeln eines Testprogramms	165
5.3.2	Modifizieren der Business-Objekt-Klasse	166
5.3.3	Testen der Änderungen	169

6 Modellklasse 171

6.1	Klassenübersicht	171
6.1.1	Was ist MVC?	172
6.1.2	Erstellen der Modellklasse	172
6.2	Transaktionale Methoden	176
6.2.1	Lesemethode	177
6.2.2	Methoden zum Sperren von Datensätzen	179
6.2.3	Speichermethode	180
6.2.4	Getter-Methoden	182
6.3	Utility-Methoden	184
6.4	Versand von E-Mails	186
6.4.1	Einrichten des E-Mail-Versands	186
6.4.2	Ausführen eines E-Mail-Tests	191
6.4.3	E-Mail-Methode	192
6.5	Reguläre Ausdrücke	198

7 ABAP und SAP NetWeaver Master Data Management 201

7.1	Was ist Master Data Management?	202
7.1.1	Technische Architektur	203
7.1.2	Clients	204
7.1.3	Grundlagen der MDM-Administration	207
7.1.4	Modellieren in SAP MDM	209
7.2	Konfigurieren der SAP MDM ABAP API	210
7.2.1	Installieren des MDM API Add-ons	210
7.2.2	Konfigurieren der MDM-API-Verbindung	213
7.2.3	Authentifizierung mit der MDM API	214
7.3	Erstellen von Code mit der MDM ABAP API	215
7.3.1	Klassenbasierte API	215
7.3.2	Funktionsbausteinbasierte API	218
7.3.3	Nicht-Unicode-ABAP-Systeme	220

- 7.3.4 Einfaches Auslesen 220
- 7.3.5 Vollständiges Auslesen 224
- 7.3.6 Integrieren des MDM Repositorys in die Business-Objekt-Klasse 226

8 ABAP Unit 229

- 8.1 Übersicht über ABAP-Unit-Tests 229
 - 8.1.1 Testklassen 230
 - 8.1.2 Testattribute 231
 - 8.1.3 Testmethoden 233
 - 8.1.4 Assertions 234
- 8.2 Erstellen von ABAP-Unit-Tests 235
 - 8.2.1 Erstellen der ABAP-Unit-Testklassen 236
 - 8.2.2 Fixture-Implementierung 237
 - 8.2.3 Implementierungen von Testmethoden 237
- 8.3 Ausführen des ABAP-Unit-Tests 240

9 Veröffentlichen eines Modells als Webservice 243

- 9.1 Webservice-Definition 243
 - 9.1.1 Service-Definitions-Wizard 244
 - 9.1.2 Service-Definition 247
 - 9.1.3 Freigeben des Webservice 249
- 9.2 Testen des Webservice 250
 - 9.2.1 Webservice-Administration 251
 - 9.2.2 Verwenden der Webservice-Homepage 252

10 Veröffentlichen eines Modells als Webservice unter Verwendung von SAP NetWeaver Process Integration 257

- 10.1 Modellieren eines Service in SAP NetWeaver Process Integration 258
 - 10.1.1 Integration Builder 259
 - 10.1.2 Einfache Datentypen 262
 - 10.1.3 Komplexe Datentypen 267
 - 10.1.4 Message-Typen 269
 - 10.1.5 Message-Interface 270
- 10.2 Implementierung des Service als Serverproxy 272
- 10.3 Erstellen einer Service-Definition 277

11 Klassisches Dynpro-UI und ALV-Objektmodell 281

11.1	Übersicht über das ALV-Objektmodell	282
11.1.1	Übersicht über die ALV-Werkzeuge	282
11.1.2	Darstellungsarten	284
11.2	Erste Schritte	285
11.2.1	Paketauswahl	285
11.2.2	Grundlegendes Programm-Coding	286
11.3	Ändern der ALV-Ausgabe	290
11.3.1	ALV-Funktionen	291
11.3.2	Bearbeiten von Spaltenattributen	292
11.3.3	Ändern der Darstellungseinstellungen	294
11.3.4	Speichern von Layout-Varianten	295
11.4	ALV in einer interaktiven Anwendung	296
11.4.1	Hinzufügen von Buttons	297
11.4.2	Definieren von Markierungen	298
11.4.3	Ereignisbehandlung	299
11.5	Objektorientierter Ansatz mit einem klassischen Dynpro	303
11.5.1	Neustrukturieren des Dialogprogramms	304
11.5.2	Erstellen der Controller-Klasse	306
11.5.3	Enjoy-Control-Ereignisse	308
11.5.4	Dynpro-Ereignisse	310

12 Web Dynpro ABAP 313

12.1	Übersicht über die Components	314
12.1.1	Component für den Kurs-Frame	315
12.1.2	Component für Fakultätsdetails	316
12.1.3	Component für die Fakultätsliste	317
12.1.4	Component für Kursdetails	318
12.2	Component-Verwendung	319
12.3	Allgemeine UI-Funktionen	321
12.3.1	Wertehilfe	322
12.3.2	Mussfelder	324
12.3.3	Änderungs- bzw. Anzeigemodus	328
12.4	ALV Component	330
12.4.1	ALV-Component-Verwendung	331
12.4.2	Context Mapping	331
12.4.3	ALV-Implementierung	333
12.5	Popup-Dialogfenster	336
12.5.1	Windows aus derselben Component	336

- 12.5.2 Windows aus einer externen Component-Verwendung 339
- 12.6 Tabellen-Popins 343
 - 12.6.1 Was ist ein Tabellen-Popin? 343
 - 12.6.2 Entwerfen von Tabellen-Popins 344
 - 12.6.3 Context Design 346
 - 12.6.4 Logik zum Füllen des Contexts für ein Popin 347
- 12.7 Hoch- bzw. Herunterladen von Dateien 348
 - 12.7.1 Herunterladen von Dateien 348
 - 12.7.2 Hochladen von Dateien 350
- 12.8 Web Dynpro Debugger 352

13 Business Server Pages 355

- 13.1 Internetdarstellung der BSP-Anwendung 356
 - 13.1.1 Zustandslos im Vergleich zu zustandsbehaftet 356
 - 13.1.2 Anwendungs-Layout 359
 - 13.1.3 Kundenspezifische Stylesheets 360
 - 13.1.4 Kursübersichtsseite 364
 - 13.1.5 AJAX 367
- 13.2 BSP-Extensions 374
 - 13.2.1 Hochladen der Open-Source-Lösung 375
 - 13.2.2 Erstellen der BSP-Extension 376
 - 13.2.3 Erstellen des BSP-Extension-Elements 377
 - 13.2.4 Designzeit-Validierung 380
 - 13.2.5 Laufzeitvalidierung 381
 - 13.2.6 Element-Rendering 382
 - 13.2.7 Testen der neuen Extension 384

14 Adobe Forms 387

- 14.1 Infrastruktur und Einrichtung 388
 - 14.1.1 Adobe Document Services-Infrastruktur 388
 - 14.1.2 Veröffentlichen des Service in der J2EE Engine 389
 - 14.1.3 Konfigurieren der Service-Schnittstellen in ABAP 390
- 14.2 Funktionsbausteinbasierte Formulare 393
 - 14.2.1 Erstellen der Schnittstelle 394
 - 14.2.2 Formularschnittstelle zum Context Mapping 395
 - 14.2.3 Formular-Layout-Editor 400
 - 14.2.4 Formularfunktionsbaustein und Kodierung 403
- 14.3 Web-Dynpro-basierte Formulare 408
 - 14.3.1 Erstellen von Web-Dynpro-Views 409

14.3.2	Formulardesign in Web Dynpro	410
14.3.3	Tabellenausgabe in Formularen	412
14.3.4	Hinzufügen von Interaktivität	415
14.4	Offline-Formulare	416
15 SAP NetWeaver Portal		421
15.1	Erstellen der Systemkonfiguration	421
15.1.1	Benutzerzuordnung	428
15.1.2	Systemtest	429
15.2	Erstellen von Portal Content	430
15.2.1	iView-Erstellung	431
15.2.2	Rollenzuweisung	434
15.2.3	Ausführen von Beispielen	436
15.3	Portal Eventing	439
15.3.1	Auslösen eines Portal-Ereignisses	440
15.3.2	Abfangen eines Portal-Ereignisses	442
15.4	SAP NetWeaver Visual Composer	445
15.4.1	Verwenden eines Webservice	446
15.4.2	Verwenden eines RFCs	448
15.4.3	Erstellen der Benutzeroberfläche	451
15.4.4	Erstellen der Wertehilfe	452
15.5	SAP NetWeaver Business Client	459
16 RSS-Feed unter Verwendung eines ICF-Serviceknotens		465
16.1	Was ist ein ICF-Service-Knoten?	466
16.2	Handler für RSS-Feeds	467
16.2.1	Testimplementierung der HTTP-Handler-Klasse	468
16.2.2	ICF-Knotenerstellung und Handler-Zuordnung	469
16.2.3	Implementierung des RSS-Handlers	470
16.2.4	Anwendungslogik des RSS-Handlers	471
16.2.5	Handler für das Herunterladen von Anhängen	477
17 Schlussbetrachtung		481
Anhang		483
A	Code-Beispiele	485
B	Die Autoren	489
Index		491