

Inhalt

| | |
|---|-----------|
| Vorwort | 11 |
| Einleitung | 13 |
| 1 Die ersten Schritte – »Hallo ADT« | 21 |
| 1.1 Backend-System | 21 |
| 1.1.1 Den SAP NetWeaver Application Server einrichten | 22 |
| 1.1.2 Amazon Web Services | 27 |
| 1.2 Installation und Einrichtung der Entwicklungsumgebung | 52 |
| 1.2.1 Java Runtime Environment | 52 |
| 1.2.2 Eclipse | 53 |
| 1.2.3 ABAP Development Tools | 54 |
| 1.3 Eclipse Workbench | 58 |
| 1.3.1 Workspaces | 58 |
| 1.3.2 Projekte | 59 |
| 1.3.3 Views | 62 |
| 1.3.4 Perspektiven | 62 |
| 1.4 Das erste Programm | 63 |
| 1.4.1 ABAP-Projekt anlegen | 63 |
| 1.4.2 »Hallo Welt« | 66 |
| 1.4.3 Feature Explorer | 69 |
| 2 Funktionen der ABAP Development Tools | 71 |
| 2.1 Demoszenario | 71 |
| 2.2 Mit Entwicklungsobjekten arbeiten | 73 |
| 2.2.1 Pakete anlegen | 74 |
| 2.2.2 ABAP-Dictionary-Objekte anlegen | 77 |
| 2.2.3 Funktionsbausteine anlegen | 82 |
| 2.2.4 Klassen | 86 |
| 2.2.5 ABAP-Programme erstellen | 92 |
| 2.3 Code-Vervollständigung | 96 |
| 2.3.1 Einfache Code-Vervollständigung | 97 |
| 2.3.2 Code-Vorlagen | 98 |
| 2.4 Quelltext-Editor | 104 |
| 2.4.1 Aufbau | 105 |
| 2.4.2 Lokale Änderungshistorie | 108 |

- 2.4.3 Lineale und Markierungen 109
- 2.4.4 Navigation 112
- 2.4.5 Textbearbeitung 114
- 2.4.6 Quelltext formatieren 117
- 2.5 Mit mehreren Ressourcen arbeiten 118
- 2.6 Verwendungsnachweis 124
- 2.7 Quick Fix und Quick Assist 126
 - 2.7.1 Quick Fix 127
 - 2.7.2 Quick Assist 134
- 2.8 Arbeitsvorrat organisieren 137
 - 2.8.1 Objekte finden 137
 - 2.8.2 Lesezeichen 143
 - 2.8.3 Aufgaben verwalten 145
 - 2.8.4 Quellcode-Links 147
 - 2.8.5 Aufgabenverwaltung mit Mylyn 150
- 2.9 Dokumentation 153
 - 2.9.1 Objekte dokumentieren 153
 - 2.9.2 Dokumentation anzeigen 156
 - 2.9.3 Dokumentation importieren 158
 - 2.9.4 ABAP Language Help 160

3 Wenn mal etwas schiefläuft: Fehler finden, analysieren und beheben 161

- 3.1 Testen 162
 - 3.1.1 ABAP-Testklassen 162
 - 3.1.2 Code Coverage 169
 - 3.1.3 ABAP Test Cockpit 172
 - 3.1.4 Laufzeitanalyse 179
- 3.2 Debugging 181
 - 3.2.1 Breakpoints setzen 182
 - 3.2.2 Debugger 183
- 3.3 Systemmeldungen abonnieren 189
- 3.4 Refactoring 192

4 Web Dynpro ABAP in Eclipse 199

- 4.1 Controller-Programmierung 199
 - 4.1.1 Kontext-Editor 202
 - 4.1.2 Attribute 206
 - 4.1.3 Methoden 206

- 4.2 Views und Windows 214
 - 4.2.1 Views 214
 - 4.2.2 Windows 229
- 4.3 Web-Dynpro-Applikationen 229
- 4.4 Floorplan Manager 231

5 Datenbankzugriffe 235

- 5.1 Data Preview 235
- 5.2 SQL-Konsole 239
- 5.3 Core Data Services 242

6 Erweiterungen der ABAP Development Tools 249

- 6.1 Eclipse Rich Client Platform 250
 - 6.1.1 Plug-in-System in Eclipse 251
 - 6.1.2 Anatomie eines Plug-ins 251
- 6.2 Server-Komponente der ABAP Development Tools 254
- 6.3 Entwicklungsumgebung für Erweiterungen vorbereiten 256
- 6.4 Implementierungen im Backend (SAP NetWeaver AS ABAP) 258
 - 6.4.1 Resource Controller 259
 - 6.4.2 Resource-Application-Klasse 263
- 6.5 Implementierungen im Frontend (Eclipse) 267
 - 6.5.1 Abhängigkeiten 271
 - 6.5.2 Verbindung zum Backend 274
 - 6.5.3 Datenaustausch 279
- 6.6 Discovery Service 285

Anhang 289

- A Wegweiser durch die ABAP Development Tools 291
- B Literatur und weiterführende Informationsquellen 299
- C Autor 303

- Index 305