

Inhaltsverzeichnis

	Einführung	1
Teil I	Die Eclipse-Rich-Client-Plattform	9
1	Architektur der Eclipse-Rich-Client-Plattform	11
1.1	Die Rich-Client-Plattform	11
1.2	Die RCP-Binärdistribution	13
1.3	Architektur eines Rich-Client-Frameworks	14
2	Modelle und Muster	19
2.1	Projektplaner	19
2.2	Eine minimale RCP-Anwendung	21
2.2.1	Die Java-Entwicklungsumgebung	22
2.2.2	Die Plugin-Development-Perspektive	22
2.2.3	Ein Plugin-Projekt anlegen	22
2.2.4	Die Anwendung testen	24
2.3	Modularisierung von Rich-Client-Anwendungen	24
2.4	Entwurfsmuster	26
2.5	Domänenmodell und Kommandos	28
2.5.1	Das Datenmodell	29
2.5.2	Die Operationen	39
2.6	Eclipse Data Binding	55
2.6.1	Grundlagen	55
2.6.2	Java Beans	57
2.6.3	Anwendung	60
2.7	Aufgabe	64

3	Plugin-Entwicklung	65
3.1	Eclipse und OSGi	65
3.2	Die Anatomie eines Plugins	66
3.2.1	Das OSGi-Manifest	66
3.2.2	Der Aktivator	70
3.2.3	Der Bundle-ClassLoader	71
3.2.4	Dienste	73
3.2.5	Erweiterungen und Erweiterungspunkte	81
3.2.6	Der Schema-Editor	83
3.3	Die Kernklassen der Eclipse-Plattform	87
3.3.1	Pfade, URLs und Dateien	88
3.3.2	Lang laufende Prozesse überwachen	89
3.3.3	Nebenläufigkeit	89
3.3.4	Weitere Basistypen	94
3.3.5	Das Eclipse-Laufzeitsystem	94
3.3.6	Die Erweiterungsregistratur	95
3.3.7	Benutzereinstellungen	99
3.3.8	Aufgabe	102
3.4	Die Eclipse-Plattform erweitern	102
3.4.1	Ausgewählte Erweiterungspunkte	102
3.4.2	Beispiel	109
3.5	Die Benutzeroberfläche der Eclipse-Plattform	112
3.5.1	Die Eclipse-Workbench	112
3.5.2	Die Architektur der Workbench	113
3.5.3	Ablaufumgebungen	117
3.5.4	Ereignisverarbeitung in der Workbench	117
3.5.5	Operationen	119
3.5.6	Weitere zentrale Dienste der Workbench	121
3.5.7	Editoren	122
3.5.8	Views	124
3.5.9	Aktionen	126
3.5.10	Dialoge	127
3.5.11	Workbench-Wizards	128
3.5.12	Präferenzen-GUI	128
3.6	Benutzerunterstützung	130
3.6.1	Das Hilfesystem	130
3.6.2	Cheat Sheets	137

4	RCP-Entwicklung	139
4.1	Plugins und die RCP	139
4.2	Applikationen	140
4.2.1	Das Interface IPlatformRunnable	140
4.3	Die Workbench der Beispielanwendung	142
4.3.1	Die Workbench konfigurieren	143
4.3.2	Workbench-Fenster konfigurieren	147
4.3.3	Aktionsleisten erzeugen	150
4.3.4	Das Anwendungs-Plugin der Beispielanwendung	155
4.3.5	Montieren einer RCP-Anwendung	160
4.3.6	Aufgabe	161
5	Produktanpassung	163
5.1	Produktgestaltung	163
5.1.1	Produktdefinition im Plugin-Manifest	164
5.1.2	Die Produktkonfiguration	164
5.1.3	Konfiguration benötigter Plugins	172
5.2	Benutzerunterstützung	174
5.2.1	Die Willkommenseite	174
5.2.2	Das Hilfesystem	178
5.3	Produkte internationalisieren	178
5.3.1	Textkonstanten in Programmen	178
5.3.2	Textkonstanten in Manifest-Dateien	180
5.3.3	Hilfetexte, Cheat Sheets und Willkommenseiten internationalisieren	182
5.3.4	Andere Ressourcen internationalisieren	182
5.3.5	Plattformkomponenten internationalisieren ...	183
5.4	Aufgabe	183
6	Eine RCP-Anwendung installieren	185
6.1	Plugins und Fragmente	185
6.2	Features	186
6.2.1	Das Feature-Manifest	186
6.2.2	Sprachpakete als Features ausliefern	190
6.2.3	Auslieferung für heterogene Plattformen	191
6.2.4	Feature-basierte Produktkonfiguration	192
6.2.5	Aufgabe	194
6.3	Build-Prozess	194
6.4	Die Update-Site	196

6.5	Konfiguration einer Rich-Client-Applikation	197
6.6	Ausliefern einer Rich-Client-Applikation	199
6.6.1	Auslieferung als ZIP-Datei	199
6.6.2	Installationshilfen	199
6.6.3	Java Web Start	200
6.7	Remote Debugging einer RCP-Anwendung	203
7	Eine RCP-Anwendung im Feld aktualisieren	205
7.1	Voraussetzungen	205
7.2	Der Eclipse-Update-Manager	206
7.3	Batch-Update	206
7.4	Programmgesteuerte Updates	208
7.5	Die Beispielanwendung aktualisieren	211
7.5.1	Plugin-Manifest	211
7.5.2	Die Klasse UpdateAction	212
7.5.3	Test der Update-Funktionalität	222
7.5.4	Automatischer Update während des Starts	223
7.6	Verwaltung der Konfiguration	224
7.7	Wenn etwas schiefgeht	225
7.8	Der Feature-Patch	225
7.9	Sicherheitsfragen	226
7.10	Alternative Update-Methoden	227
8	RCP-Kochbuch	229
Teil II	Benutzeroberflächen für Rich Clients	231
9	SWT, JFace und das Forms API	233
9.1	Das Standard Widget Toolkit	234
9.1.1	Übersicht über die SWT-Funktionsgruppen ...	234
9.1.2	Ereignisse	235
9.1.3	Widgets	236
9.1.4	Layouts	241
9.1.5	SWT-Dialoge	244
9.1.6	Grafik	245
9.1.7	Texte	248
9.1.8	Der Mauszeiger	249
9.1.9	Swing-Integration	249
9.1.10	Druckerunterstützung	250
9.1.11	Datentransfer	255
9.1.12	Ressourcenverwaltung	257

9.1.13	Windows32-Unterstützung (OLE)	257
9.1.14	OpenGL	258
9.1.15	Eigene Widgets	258
9.1.16	Aufgabe	262
9.2	JFace	262
9.2.1	Ressourcenverwaltung	262
9.2.2	Dialoge und Fenster	264
9.2.3	Viewer	269
9.2.4	Textverarbeitung	272
9.2.5	Inhaltsassistenten für Felder	279
9.2.6	Aktionen und Menüs	279
9.2.7	Wizards	281
9.2.8	Präferenzen	282
9.2.9	Das JFace-Databinding	284
9.3	Das Forms API	287
9.3.1	Grundlagen	287
9.3.2	FormText	290
9.3.3	Kollabierbare Behälter	291
9.3.4	Layouts	292
9.3.5	Ressourcenverwendung	292
9.3.6	Trennung von Datenmodell und Repräsentation	293
9.4	Ein GUI-Framework für die Beispielanwendung	293
9.4.1	Der MasterDetailsBlock	294
9.4.2	Die Detailseite	299
9.4.3	Die View-Komponente	317
9.4.4	Formulare ausdrucken	328
9.5	Entwurfsfragen	339
10	Jenseits von SWT	341
10.1	SWT-Erweiterungen	341
10.2	Skriptsprachen für Benutzeroberflächen	342
10.2.1	XML-Skriptsprachen	342
10.2.2	XML-GUI-Beschreibung in der Beispielanwendung	345
10.2.3	Andere Skriptsprachen	350

10.2.4	Schlussbetrachtung	355
10.2.5	Aufgabe	355
Teil III	Persistenz	357
11	Dateisysteme und Eclipse-Workspace	359
11.1	Standard-Java-IO	359
11.2	Das Eclipse-Datei-System	360
11.3	Der Eclipse-Workspace	361
11.4	Grenzen	361
12	XML	363
13	Relationale Datenbanksysteme für Rich Clients	367
13.1	Die Eclipse Data Tools Platform	368
13.2	DBMS in der Rich-Client-Anwendung	371
13.2.1	Eingebetteter Betrieb und Client/Server-Betrieb	372
13.2.2	Ein eigenes Derby-Plugin	373
13.2.3	Aufgabe	376
13.3	Transaktionen	377
13.4	Datenbankzugriff für die Beispielanwendung	382
13.4.1	Autoinstallation	382
13.4.2	Datenmodell und Datenzugriffsmodul	383
13.4.3	Benutzeranmeldung	389
13.4.4	Benutzerverwaltung	390
13.5	Zusammenfassung	399
14	Objektrelationale Brücken	401
14.1	JPA	401
14.1.1	Die JPA-Architektur	403
14.1.2	JPA in einer Rich-Client-Anwendung	405
14.1.3	Das Datenmodell annotieren	410
14.1.4	Datenbankabfrage	413
14.1.5	Objekte und Entities	413
14.1.6	Transaktionen	415
14.1.7	Aufgabe	417
14.2	Ein Plugin zur Ressourcenverwaltung	418
14.2.1	Plugin-Manifest	418
14.2.2	Die Klasse ResourceView	420
14.2.3	Die Klasse PerspectiveFactory	430
14.2.4	Test	431

14.3	Das Plugin für die Projektplanung	432
14.3.1	Der Master-Bereich	432
14.3.2	Der Detailbereich	436
14.3.3	Die Detailseiten	437
14.3.4	Der View	440
15	Andere Persistenzoptionen	445
15.1	Objektorientierte Datenbanken	447
15.1.1	Prototypen	447
15.1.2	Abfragen	448
15.1.3	Zusammenfassung	449
15.2	Prevayler	450
15.2.1	Konzept	450
15.2.2	Anwendung	451
15.2.3	Zusammenfassung	454
Teil IV	Der Rich Client wird noch reicher	455
16	Das Graphical Editing Framework	457
16.1	Stärken und Schwächen von GEF	457
16.2	Die GEF-Architektur	458
16.2.1	Draw2D	458
16.2.2	GEF aus der Vogelperspektive	461
16.3	Modell, Viewer, Controller	462
16.3.1	EditParts	462
16.3.2	Kommandos	464
16.3.3	Werkzeuge	464
16.4	Beispielanwendung: Gantt-Chart mit GEF	465
16.4.1	GEF installieren	465
16.4.2	Der Gantt-View	466
16.4.3	Aufgabe	492
16.5	Andere grafische Möglichkeiten	492
17	BIRT	495
17.1	Die Entwurfsphase	495
17.1.1	Installation	495
17.1.2	Eine Datenquelle	496
17.1.3	Data Sets	497
17.1.4	Parameter	499
17.1.5	Layout	499

17.2	BIRT in einer Rich-Client-Anwendung	501
17.2.1	Die BIRT-Architektur	501
17.2.2	Berichte in einer RCP-Anwendung generieren und anzeigen	502
17.2.3	Zusammenfassung	511
17.2.4	Aufgabe	511
18	Elefantenhochzeit: Office-Komponenten unter Eclipse	513
18.1	OpenOffice einbetten	514
18.1.1	Benötigte Komponenten	514
18.1.2	OpenOffice als Server konfigurieren	514
18.1.3	Wie OpenOffice mit Java spricht	515
18.1.4	Native Bibliotheken	516
18.1.5	Die Infrastruktur bereitstellen	516
18.2	Ein OpenOffice-View	522
18.2.1	Projekt anlegen	523
18.2.2	Die Klasse <code>SpreadsheetsView</code>	525
18.2.3	Die Klasse <code>CalcView</code>	530
18.2.4	Aufgabe	533
18.3	Office-Dateien generieren	533
19	Ausgabe von PDF-Dateien	545
19.1	PDF-Dateien erzeugen	545
19.2	Ein PDF-Dokument anzeigen	553
19.3	Aufgabe	555
Teil V	Administration	557
20	Synchronisierung mit dem Server	559
20.1	Allgemeine Aspekte der Datensynchronisierung	559
20.2	Proprietäre Lösungen	561
20.2.1	Oracle LITE	561
20.2.2	IBMs DB2 Everyplace	561
20.2.3	DaffodilReplicator	562
20.2.4	Microsoft .NET DataSets	562
20.3	Standards (OMA DS)	563
20.4	Lösungen unter Java	566
20.5	Ein synchronisierter Task-Planer	568
20.5.1	XML-Serialisierung	569
20.5.2	Das Sync-Projekt	569

21	Konfiguration und Administration	591
21.1	Benutzerauthentifizierung	591
21.2	Automatische Konfiguration	592
21.2.1	Plugins entfernen	593
21.2.2	Flexiblere Steuerung	595
21.3	Statistiken	596
21.3.1	Aufgabe	596
21.4	Fernwartung	597
22	Schlussfolgerungen	599
22.1	Workflow	599
22.2	ClassLoader	600
22.3	Programmgröße	600
22.4	Schutz intellektuellen Eigentums	601
Anhang		603
Anhang A:		
	Wichtige Download-Adressen	605
Anhang B:		
	Erste Hilfe	607
Anhang C:		
	Lösungen zu den Aufgaben	611
Anhang D:		
	Relevante eclipse.org-Projekte	621
Anhang E:		
	Mylin	623
Bibliografie		625
Index		627