

Berthold Daum

Java-Entwicklung mit Eclipse 3.3

Anwendungen, Plugins und Rich Clients

5., aktualisierte und erweiterte Auflage



dpunkt.verlag

Dr. Berthold Daum
E-Mail: berthold.daum@bdaum.de

Lektorat: René Schönfeldt
Copy-Editing: Annette Schwarz, Ditzingen
Satz und Herstellung: Birgit Bäuerlein
Umschlaggestaltung: Helmut Kraus, www.exclam.de
Druck und Bindung: Koninklijke Wöhrmann B.V., Zutphen, Niederlande

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

ISBN 978-3-89864-504-1

5., aktualisierte und erweiterte Auflage 2008
Copyright © 2008 dpunkt.verlag GmbH
Ringstraße 19
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.

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

	Einführung	1
Teil I	Das Java-IDE	9
1	Wo ist was?	11
2	Effizientes Programmieren mit Eclipse	45
3	Die Kunst der (visuellen) Komposition	77
4	Projekte richtig organisieren	85
5	Projekt 1: Duke spricht	115
6	Programmentwicklung	141
7	Weiterführende Themen der Projektentwicklung	181
Teil II	SWT statt Swing	193
8	Das SWT	195
9	JFace	271
10	Projekt 2: Jukebox	311

Teil III	Eclipse als Ablaufplattform	395
11	Plugins für die Eclipse-Workbench entwickeln	397
12	Eigene Produkte auf der Grundlage von Eclipse entwickeln	499
13	Projekt 3: Rechtschreibprüfung als Eclipse-Plugin	523
14	Die Rich Client Platform	631
15	Projekt 4: Das Spiel Hex als RCP-Applikation	651
16	Einführung in OSGi und Equinox	683
17	Projekt 5: Zeitansage	699
18	Schlussfolgerungen und Ausblick	715
	Anhang	721
A	Nützliche Plugins für Eclipse	723
B	Verhalten bei einem Eclipse-Versionswechsel	727
C	Wichtige Downloads	731
	Bibliographie	733
	Index	737

Inhaltsverzeichnis

Einführung	1
Was ist Eclipse?	1
Das Eclipse-Umfeld	3
Über dieses Buch	3
Wie dieses Buch organisiert ist	4
Danksagung	7
Teil I Das Java-IDE	9
1 Wo ist was?	11
1.1 Eclipse installieren	11
1.2 Die Plattform erweitern	17
1.2.1 Auslieferung mit einem Installationsmanager ...	17
1.2.2 Auslieferung in Form eines URL	17
1.2.3 Auslieferung als ZIP-Datei	18
1.2.4 Extension Sites	19
1.3 Die erste Applikation: Hello World	19
1.4 Die wichtigsten Präferenzen für die Java-Entwicklung ...	25
1.4.1 Workbench-Einstellungen	26
1.4.2 Installierte JREs	27
1.4.3 Compiler-Einstellungen	29
1.4.4 Codeformatierung	31
1.4.5 Templates	32
1.5 Tasks und Probleme	34
1.5.1 Probleme, Probleme	34
1.5.2 Die Aufgabenliste	37
1.5.3 Lesezeichen	39
1.6 Das Scrapbook	39

1.7	Mylin Tasks	41
1.7.1	Aufgabenorientierte Arbeitsweise	41
1.7.2	Task-Repositoryen	42
2	Effizientes Programmieren mit Eclipse	45
2.1	Kleine Helfer	45
2.1.1	Systeminformationen	45
2.1.2	Help und Hover-Info	46
2.1.3	Java-Informationsfenster	48
2.1.4	Automatische Codevervollständigung	49
2.1.5	Der Korrektur-Assistent	58
2.1.6	QuickAssist	59
2.1.7	Komfortfunktionen des Java-Editors	60
2.2	Sich im Code zurechtfinden	61
2.3	Programme neu strukturieren	63
2.3.1	Änderungen an Typen	64
2.3.2	Den Code restrukturieren	65
2.4	Undo und Redo	70
2.5	Local History	71
2.5.1	Ressourcen vergleichen	71
2.5.2	Durch ältere Version ersetzen	72
2.5.3	Gelöschte Ressource wiederherstellen	73
2.6	Unterstützung für Java 5 und Java 6	73
2.6.1	Generische Datentypen	73
2.6.2	Enumerationen und Annotationen	74
2.6.3	Autoboxing	74
2.6.4	Statische Importe	75
2.6.5	Neue Syntax für for-Schleife	75
2.6.6	Variable Anzahl von Parametern	75
2.6.7	Kategorien	76
3	Die Kunst der (visuellen) Komposition	77
3.1	Einstellungen	78
3.2	Komposition	79
3.3	Layouts	81
3.4	Ereignisverarbeitung	82

4	Projekte richtig organisieren	85
4.1	Die Workbench	85
4.2	Ressourcen	86
4.2.1	Ressourcentypen	86
4.2.2	Wo Ressourcen gespeichert werden	87
4.2.3	Metadaten und Synchronisation	88
4.2.4	Explorer-Konfiguration	89
4.2.5	Working Sets	89
4.3	Assoziationen	90
4.4	Packages	93
4.4.1	Verzeichnisse und Packages	93
4.4.2	Navigation	93
4.4.3	Typhierarchie	94
4.5	Der Outline-View	95
4.5.1	Darstellung	96
4.5.2	Kontextfunktionen	97
4.6	Suchen	99
4.6.1	Die Suchfunktion	99
4.6.2	Finden und Ersetzen	101
4.6.3	Gleichnamige Elemente markieren	102
4.7	Editoren und Views anordnen	103
4.8	Perspektiven verwalten	105
4.8.1	Neue Perspektiven definieren	105
4.8.2	Perspektiven konfigurieren	106
4.8.3	Mehrere Perspektiven öffnen	107
4.9	Dateien importieren	107
4.10	Der Java Build Path	109
4.10.1	JARs im Build Path	110
4.10.2	Bibliothekskonfiguration	112
4.11	Die Java-Browsing-Perspektive	113
5	Projekt 1: Duke spricht	115
5.1	Einrichten des Projekts	115
5.2	Kurzer Exkurs in die Sprachsynthese	116
5.3	Erweiterung des FreeTTS-Systems	117
5.3.1	Animationseignisse	118
5.3.2	Der Animator	119
5.3.3	Einbau in FreeTTS	123
5.3.4	Verbindung mit dem Java-Audiosystem	126

5.4	Die Benutzeroberfläche	128
5.4.1	Das animierte Gesicht	129
5.4.2	Die Bedieneinheit	129
5.4.3	Das Modell	130
5.4.4	Die Präsentation	130
5.4.5	Die komplette Applikation	135
5.5	Die Applikation aus Eclipse exportieren	138
5.6	Vertiefende Literatur	139
5.7	Was konnten wir lernen?	140
6	Programmentwicklung	141
6.1	Debugging	141
6.1.1	Die Debug-Perspektive	141
6.1.2	Die Debug-Konfiguration	143
6.1.3	Den Programmablauf steuern	144
6.1.4	Breakpoints verwalten	147
6.1.5	Die Konsole	149
6.1.6	Remote-Debugging	149
6.2	JUnit	150
6.2.1	JUnit einrichten	151
6.2.2	Testfälle erstellen	153
6.2.3	Testfälle zum Ablauf bringen	154
6.3	Die Eclipse Test and Performance Tools Platform (TPTP)	155
6.3.1	Statische Codeanalyse	156
6.3.2	Testen	157
6.3.3	Performanzmessungen	163
6.3.4	Überwachung	169
6.4	Dokumentation	173
6.5	Ant	176
6.5.1	Ant konfigurieren	177
6.5.2	Import und Export	177
6.5.3	Ein Ant-Skript editieren	178
6.5.4	Ein Ant-Skript ausführen	178
6.5.5	Ant-Skripte in Build-Prozesse integrieren	180
7	Weiterführende Themen der Projektentwicklung	181
7.1	Entwicklung im Team	181
7.1.1	Repository einrichten	182
7.1.2	Projekte im Repository	184

7.1.3	Versionsverwaltung	185
7.1.4	Arbeiten im Team	186
7.1.5	Weitere CVS-bezogene Funktionen	188
7.2	Andere Quellcodeverwaltungssysteme	189
7.3	Weitere Möglichkeiten zum Austausch von Entwicklungsobjekten	189
7.4	Externe Werkzeuge	191

Teil II SWT statt Swing 193

8	Das SWT	195
8.1	Übersicht über die SWT-Funktionsgruppen	195
8.2	Vor- und Nachteile des SWT	196
8.2.1	Vorteile des SWT	196
8.2.2	Nachteile des SWT	197
8.2.3	Interoperabilität	198
8.3	Das Package SWT	198
8.4	Ereignisse	199
8.4.1	Zuhörer	199
8.4.2	Adapter	200
8.4.3	Ereignisse	201
8.4.4	Übersicht über Listener-, Adapter- und Event-Klassen	201
8.5	Widgets	203
8.5.1	Visuelle Übersicht	205
8.5.2	Displays, Monitore, Shells und ToolTips	206
8.5.3	Dialoge	213
8.5.4	Composites, Groups und Canvas	215
8.5.5	Tasten	217
8.5.6	Schieberegler, Skalen und Fortschrittsbalken ..	218
8.5.7	Text- und numerische Felder, Beschriftungen und Verweise	220
8.5.8	Tabellen, Listen und Combos	223
8.5.9	Bäume	229
8.5.10	Verschiebbare Trennleisten	231
8.5.11	Pultordner	231
8.5.12	Werkzeugleisten und Menüs	233
8.5.13	Nicht native Widgets	239
8.5.14	Das Browser-Widget	241

8.6	Layout-Manager	242
8.6.1	Visuelle Übersicht	243
8.6.2	Die Klasse FillLayout	244
8.6.3	Die Klasse RowLayout	244
8.6.4	Die Klasse GridLayout	245
8.6.5	Die Klasse FormLayout	248
8.6.6	Die Klasse StackLayout	250
8.7	Grafik	251
8.7.1	Der Grafikkontext	251
8.7.2	Farben	252
8.7.3	Schriftarten	253
8.7.4	Bilder	254
8.7.5	Texte	255
8.7.6	Der Mauszeiger	256
8.7.7	Fortgeschrittene Grafikfunktionen	256
8.8	Ein Widget mit Swing	257
8.8.1	Eingebettete Inhalte	258
8.8.2	Ereignisse	258
8.9	Ausgabe auf dem Drucker	261
8.10	Datentransfer	264
8.10.1	Die Zwischenablage	264
8.10.2	Drag&Drop	265
8.11	Ressourcenverwaltung	266
8.12	Windows-32-Unterstützung (OLE)	267
8.13	OpenGL und SWT	267
8.14	SWT auf dem Pocket PC	268
8.15	Barrierefreie Software	269
9	JFace	271
9.1	Ressourcenverwaltung	271
9.1.1	Farbverwaltung	271
9.1.2	Die Klasse FontRegistry	272
9.1.3	Die Klasse ImageRegistry	272
9.1.4	Die Klasse JFaceResources	272
9.2	Dialoge und Fenster	273
9.2.1	Verschiedene Dialogtypen	274
9.2.2	Eigene Dialoge implementieren	277
9.2.3	Dialoge persistent machen	281

9.3	Viewer	282
9.3.1	Das Viewer-Ereignismodell	283
9.3.2	Die Viewer-Hierarchie	283
9.3.3	Zelleneditoren	286
9.3.4	Datentransfer	287
9.4	Textverarbeitung	288
9.4.1	Basisklassen der Textverarbeitung	288
9.4.2	Der SourceViewer	294
9.4.3	Der ProjectionViewer	298
9.5	Aktionen und Menüs	298
9.5.1	Das IAction-Interface	299
9.5.2	Die Manager	300
9.6	Wizards	301
9.6.1	Die Klasse Wizard	301
9.6.2	Die Klasse WizardPage	302
9.6.3	Die Klasse WizardSelectionPage	303
9.6.4	Die Klasse WizardDialog	303
9.7	Präferenzen	304
9.7.1	Die Klassen PreferenceStore und PreferenceConverter	304
9.7.2	Die Klasse PreferencePage	305
9.7.3	Feldeditoren	306
9.7.4	Präferenzenbäume	307
9.8	Databinding	309
9.8.1	Eclipse Databinding	309
9.8.2	JFace Databinding	310
10	Projekt 2: Jukebox	311
10.1	Designziele und Umsetzung	311
10.2	Einrichten des Projekts	314
10.3	Das Player-Modul	316
10.4	Das Datenmodell der Playlist	344
10.5	Das Fenster für die Anzeige der Beschreibung	347
10.6	Der Playlist-Viewer	349
10.7	Der Editor für Beschreibungen	380
10.8	Die Jukebox zum Einsatz bringen	392
10.9	Diskussion	393

Teil III Eclipse als Ablaufplattform		395
11	Plugins für die Eclipse-Workbench entwickeln	397
11.1	Die Architektur der Eclipse-Plattform	398
11.1.1	OSGi	398
11.1.2	Minimalplattform	400
11.1.3	Rich Client Platform vs. IDE	400
11.1.4	Ressourcenverwaltung	401
11.1.5	Benutzeroberfläche	402
11.1.6	Hilfesystem	402
11.1.7	Unterstützung für Arbeitsgruppen	402
11.1.8	Weitere Plugin-Gruppen	403
11.1.9	Zusammenfassung	403
11.2	Die Kernklassen der Eclipse-Plattform	403
11.2.1	Die Klasse Platform	404
11.2.2	Die Klasse Plugin	405
11.2.3	Die Klasse Preferences	406
11.2.4	Pfade und Dateiorde	407
11.2.5	Überwachung lang laufender Prozesse	407
11.2.6	Nebenläufigkeit	408
11.2.7	Adapter	408
11.3	Der Eclipse-Workspace	409
11.3.1	Das Eclipse-Dateisystem	409
11.3.2	Workspace-Ressourcen	410
11.3.3	Markierungen	414
11.3.4	Auf Ressourcenänderungen reagieren	415
11.3.5	Lang laufende Prozesse verwalten	417
11.4	Plugins konfigurieren	418
11.4.1	Die Plugin-Development-Perspektive	418
11.4.2	Die Manifestdateien	420
11.4.3	Die wichtigsten Erweiterungspunkte im Eclipse-SDK	426
11.4.4	Der Schemaeditor	434
11.5	Die Benutzeroberfläche der Eclipse-Plattform	437
11.5.1	Formulare	438
11.5.2	Die Eclipse-Workbench	445
11.5.3	Die Architektur der Workbench	446
11.5.4	Ereignisverarbeitung in der Workbench	448
11.5.5	Operationenverwaltung	451
11.5.6	Editoren	452

11.5.7	Views	461
11.5.8	Aktionen	466
11.5.9	Dialoge	472
11.5.10	Workbench-Wizards	475
11.5.11	Präferenzen und Properties-Seiten	478
11.5.12	Perspektiven definieren	479
11.6	Elemente der Benutzerhilfe	481
11.6.1	Das Hilfesystem	481
11.6.2	Cheat Sheets	489
11.6.3	Den Eclipse-SDK-Willkommensschirm erweitern	493
11.6.4	Hover	495
11.6.5	Mehr Komfort für GUI-Elemente	496
11.7	Webbrowser aufrufen	497
12	Eigene Produkte auf der Grundlage von Eclipse entwickeln	499
12.1	Ant-Einbindung	500
12.2	Plugins und Fragmente	500
12.3	Features	501
12.4	Deployment	504
12.4.1	Features ausliefern	504
12.4.2	Eigenständige Produkte ausliefern	505
12.4.3	Produktgestaltung	507
12.4.4	Den Workspace bevölkern	508
12.4.5	Update-Sites erstellen	511
12.5	Von einer Update-Site installieren	513
12.5.1	Update-Site hinzufügen	513
12.5.2	Features installieren	513
12.5.3	Features aktualisieren	514
12.5.4	Die Eclipse-Konfiguration verwalten	514
12.5.5	Install-Handler	515
12.5.6	Update Policies	516
12.6	Produkte internationalisieren	516
12.6.1	Textkonstanten in Programmen	517
12.6.2	Textkonstanten in Manifestdateien	519
12.6.3	Hilfetexte und Spickzettel	520
12.6.4	Sprachpakete ausliefern	521
12.7	Der Feature-Patch	521

13	Projekt 3: Rechtschreibprüfung als Eclipse-Plugin	523
13.1	Die Kernklassen der Rechtschreibprüfung	524
13.2	Übersicht	524
13.2.1	Die Engine	524
13.2.2	Übersicht	525
13.3	Das Projekt einrichten	526
13.4	Die Plugin-Konfiguration	529
13.4.1	Importierte Dateien	530
13.4.2	Das OSGi-Manifest MANIFEST.MF	530
13.4.3	Das Plugin-Manifest plugin.xml	532
13.4.4	Das Schema documentTokenizer.exsd	537
13.5	Die Aktivator-Klasse	539
13.6	Die Aktion »Check Spelling«	544
13.6.1	Die Klasse SpellCheckingTarget	544
13.6.2	Die Klasse CheckSpellingActionDelegate	551
13.6.3	Die Klasse SpellCheckingController	553
13.6.4	Die Klasse SpellCheckingJob	560
13.7	Das Korrekturfenster	562
13.7.1	Die Klasse SpellCorrectionView	562
13.7.2	Aktionen	575
13.7.3	Bilder verwalten	576
13.8	Das Zusammenspiel zwischen den Kernklassen und den GUI-Klassen	578
13.9	Dokumente analysieren	591
13.10	Die Rechtschreibprüfung konfigurieren	592
13.10.1	Präferenzen	592
13.10.2	Das Präferenzen-GUI	595
13.10.3	Den PreferenceStore auslesen	599
13.11	Das Hilfesystem	600
13.11.1	Das Hilfe-Inhaltsverzeichnis	600
13.11.2	Kontextabhängige Hilfe	601
13.11.3	Aktive Hilfe	602
13.11.4	Dynamische Hilfeseiten	603
13.12	Ein Plugin für Java-Properties	604
13.12.1	Das Projekt einrichten	605
13.12.2	Das Manifest	605
13.12.3	Die Aktivator-Klasse	608
13.12.4	Die Präferenzen	608
13.12.5	Die PreferencePage	609

13.12.6	Der Java-Properties-Tokenizer	611
13.12.7	Das Hilfesystem	611
13.13	Die Rechtschreibprüfung internationalisieren	612
13.13.1	Textkonstanten in Java-Code	612
13.13.2	Textkonstanten in Manifestdateien	616
13.13.3	Ein Sprachfragment erstellen	616
13.14	Die Rechtschreibprüfung zum Einsatz bringen	620
13.14.1	Definition des SpellChecker-Features	620
13.14.2	Das Sprach-Feature erstellen	625
13.14.3	Definition der Update-Site	627
13.14.4	Installation	629
13.15	Erfahrungen	629
14	Die Rich Client Platform	631
14.1	Motivation und Abgrenzung	631
14.2	Plugins und die RCP	632
14.3	Erstellen einer Applikation	633
14.3.1	Das Interface IApplication	634
14.3.2	Die Klasse WorkbenchAdvisor	635
14.3.3	Die Klasse WorkbenchWindowAdvisor	636
14.3.4	Die Klasse ActionBarAdvisor	637
14.4	Testen einer Rich-Client-Applikation	638
14.5	Produktdefinition	639
14.5.1	Produktdefinition im Plugin-Manifest	639
14.5.2	Die Konfigurationsdatei config.ini	640
14.5.3	Der Produkteditor	640
14.5.4	Weitere Gestaltungsmöglichkeiten	645
14.6	Die Willkommenseite	646
14.6.1	Die Willkommenseite selbst implementieren ..	646
14.6.2	Universal Intro benutzen	646
14.7	Ausliefern einer Rich-Client-Applikation	649
15	Projekt 4: Das Spiel Hex als RCP-Applikation	651
15.1	Übersicht	651
15.2	Das Projekt einrichten	653
15.3	Das Manifest plugin.xml	654
15.3.1	Erster Test	655
15.3.2	Weitere benötigte Eclipse-Plugins	655
15.3.3	Die Applikation konfigurieren	655

15.4	Die Schnittstelle zur Game-Engine	657
15.4.1	Das Interface IStatusListener	657
15.4.2	Das Interface IGame	657
15.5	Die Klasse View	658
15.6	Die Game-Engine	660
15.7	Die Klasse ApplicationActionBarAdvisor	666
15.8	Benutzerhilfe	670
15.8.1	Hilfe anbinden	670
15.8.2	Willkommensschirm erstellen	671
15.9	Produktdefinition	675
15.10	Das komplette Manifest	678
15.11	Auslieferung	680
15.12	Diskussion	681
16	Einführung in OSGi und Equinox	683
16.1	OSGi-Grundlagen	683
16.1.1	OSGi-Spezifikationen	684
16.1.2	OSGi-Bündel	685
16.1.3	Dienste	686
16.2	Die Eclipse-Equinox-Implementierung	689
16.2.1	OSGi-Standard-Packages	689
16.2.2	Zusätzlich Equinox-Bundles	690
16.3	Persistenz in Equinox-Anwendungen	691
16.4	Zugriff auf Bundle-Ressourcen	692
16.5	Benutzeroberflächen	693
16.6	Nebenläufigkeit	695
16.7	Eine OSGi-/Equinox-Anwendung starten	696
16.8	Eine OSGi-/Equinox-Anwendung ausliefern	697
17	Projekt 5: Zeitansage	699
17.1	Die Architektur der Anwendung	699
17.2	Das Projekt einrichten	699
17.3	Ausführen der Anwendung	700
17.4	Die Implementierung des Chron-Servers	701
17.5	Die Implementierung des Sprachdienstes	705
17.6	Die Implementierung der Zeitansage	710
17.7	Auslieferung	713
17.8	Diskussion	714

18	Schlussfolgerungen und Ausblick	715
18.1	Lauffähige Prototypen	716
18.2	Automatisierte Tests	717
18.3	Verfeinerungen	717
18.4	»Embrace Change«	719
18.5	Entwurfsmuster	720
18.6	Weg des geringsten Widerstandes	720
Anhang		721
A	Nützliche Plugins für Eclipse	723
B	Verhalten bei einem Eclipse-Versionswechsel	727
C	Wichtige Downloads	731
	Bibliographie	733
	Index	737