

Kai Brüssau/Oliver Widder (Hrsg.)  
Herbert Brückner, Roger Butenuth, Bernd Kolb,  
Martin Lippert, Matthias Lübken, Birgit Schwartz-Reinken,  
Markus Völter, Lars Wunderlich

# Eclipse - Die Plattform

## Enterprise-Java mit Eclipse 3

Software & Support Verlag 2005

# Inhaltsverzeichnis

<b>Vorwort.....</b>	<b>15</b>
<b>1 Einleitung.....</b>	<b>17</b>
1.1 Was ist Eclipse?.....	17
1.2 Warum ein Buch über Eclipse?.....	17
1.3 An wen richtet sich dieses Buch?.....	18
1.4 Was erwartet Sie in diesem Buch?.....	18
1.5 Was benötigen Sie?.....	19
1.6 Anforderungen an eine moderne IDE.....	19
1.7 OpenSource.....	20
<b>2 Wegweiser durch Eclipse.....</b>	<b>23</b>
2.1 Perspektiven und Views von Eclipse.....	23
2.1.1 Perspektiven.....	23
2.1.2 Perspektiven auswählen.....	23
2.1.3 Die Java-Perspektive.....	24
2.1.4 Neues Java-Projekt erzeugen.....	25
2.1.5 Package Explorer und View-Funktionen.....	25
2.1.6 Das Eclipse-Editor-Fenster.....	27
2.1.7 Outline View.....	28
2.1.8 Tasks/Problems View.....	28
2.1.9 To be done - Arbeiten mit Tasks.....	29
2.1.10 Eigene Tasks definieren.....	30
2.2 Hilfe in Eclipse.....	31
2.3 Eclipse-Projekte individuell konfigurieren.....	35
2.3.1 Vorhandenen Sourcecode ins Projekt.....	38
2.3.2 Projekteigenschaften definieren.....	39
2.4 Sourcecode-Editor-Features.....	41
2.4.1 Quickfix auf Variablen.....	44
2.4.2 Automatische Getter/Setter-Erzeugung.....	45
2.4.3 Sourcecode-Vervollständigung (Code Completion).....	46

c

\*•

## Inhaltsverzeichnis

2.4.4	Try-Catch-Block-Generierung.....	47
2.4.5	Sourcecode Templates verwenden.....	49
2.4.6	Gimmicks des Editors.....	51
2.4.7	Immer ein Undo für die Vergangenheit.....	52
2.5	Refactoring mit Eclipse.....	52
2.5.1	Überschreiben/Implementieren von Methoden.....	56
2.5.2	Auslagern von Zeichenketten/Mehrsprachigkeit.....	58
2.6	Erstellung eines JUnit-Tests.....	59
2.7	Debuggen in Eclipse.....	62
2.7.1	Breakpoints.....	62
2.7.2	Aufsetzen einer Launch Configuration.....	64
2.7.3	Die Debug-Perspektive im Überblick.....	67
2.7.4	Grundfunktionen des Debuggens.....	68
2.7.5	Allgemeine Hinweise zum Debuggen in Eclipse.....	69
- 2.7.6	Debug View.....	71
2.7.7	Breakpoints, Watchpoints und Exception Catching.....	75
2.7.8	Hot Code Replacement.....	79
2.8	Navigation im Sourcecode.....	80
2.8.1	Javadoc View.....	80
2.8.2	Navigation über die Toolbar.....	81
2.8.3	Suchfunktionen in Eclipse.....	82
2.8.4	Open-Funktionen.....	84
2.9	Weitere Views.....	86
2.9.1	Hierarchy View.....	86
2.9.2	Call Hierarchy View.....	87
2.9.3	Declaration View.....	87
2.9.4	Navigator View.....	88
2.9.5	Progress View.....	88
2.10	Customizing Eclipse.....	88
2.10.1	Customizing perspective.....	89
2.10.2	Eclipse individuell starten.....	89
2.11	Eclipse Preferences.....	92

<b>Eclipse als Modellierungswerkzeug.....</b>	<b>103</b>
3.1 Einleitung ins Thema.....	103
3.2 Omondo EclipseUML.....	103
3.2.1 Versionen von EclipseUML.....	103
3.2.2 Download und Installation von Omondo EclipseUML.....	104
3.2.3 Klassendiagramme erstellen.....	105
3.3 Togetherfür Eclipse.....	110
3.3.1 Entwurfsmuster mit Together.....	115
<b>Qualitätssicherung in Eclipse.....</b>	<b>119</b>
4.1 Einführung ins Thema.....	119
4.2 Wie äußert sich ein Mangel an Qualität?.....	119
4.2.1 Mangelnde Sourcecode-Tests.....	119
4.2.2 Mangelnde Tests der fachlichen Funktionalität.....	120
4.2.3 Eingeschränkte Funktionalität.....	120
4.2.4 Begrenzte Sourcecode-Qualität.....	120
4.2.5 Reviews und Softwareunterstützung.....	121
4.3 Eclipse zur Qualitätssicherung.....	122
4.3.1 Testen mit JUnit.....	122
4.3.2 Audits mit Checkstyle.....	123
4.3.3 Sourcecode Metrics.....	126
4.3.4 Eclipse Metrics von Team-In-A-Box.....	127
<b>Eclipse und CVS.....</b>	<b>131</b>
5.1 Einführung in CVS.....	131
5.1.1 Grundlagen.....	131
5.1.2 Aufbau des Kapitels.....	133
5.2 Einrichten eines Servers.....	133
5.2.1 Linux.....	134
5.2.2 Windows.....	136
5.2.3 Nutzerverwaltung.....	137
5.3 Rundgang durch CVS.....	138
5.3.1 Verbindung zum Repository.....	138
5.3.2 Projekt im CVS ablegen.....	140

## Inhaltsverzeichnis

5.3.3	Projekt aus dem CVS auschecken.....	142
5.3.4	Mit CVS entwickeln.....	143
5.4	Tags.....	148
5.5	CVS ohne C: Locking.....	149
5.6	Branch und Merge.....	150
5.6.1	Einfacher Branch (Flying Fish).....	152
5.6.2	Probleme durch Schlüsselwörter.....	154
5.6.3	Lang laufender Branch.....	154
5.6.4	Parallele und geschachtelte Branches.....	156
5.6.5	Sorgfalt ist wichtig.....	156
5.7	Zusammenfassung.....	158

## Build Management mit Eclipse .....!.....159

6.1	Aufgaben des Build Management.....	159
6.2	Vom Quellcode zum Programm: Build.....	160
6.3	Einstellungen des Eclipse Build.....	161
6.3.1	Build-Path-Einstellungen.....	161
6.3.2	Compiler- und Javadoc-Einstellungen.....	164
6.3.3	Automatische und manuelle Builds.....	165
6.3.4	Weitere Konfigurationen.....	166
6.3.5	Erzeugung anderer Ausgabetypen.....	168
6.4	Exkurs: Build Management mit Ant.....	169
6.4.1	Ant Projects, Targets und Tasks.....	171
«6.4.2	Typische Ziele.....	173
6.4.3	Ein Ant-Beispielskript.....	174
6.4.4	Ant in Eclipse: Externe Builds.....	178

## 7

7.1	Allgemeines.....	183
7.2	Der Visual Editor für Java/Swing ..!.....	184
7.2.1	Entwicklung eines GUI mit dem Visual Editor.....	184
7.2.2	Codegenerierung.....	189
7.2.3	Beispiel.....	192
7.3	DasSWT.....	209

7.3.1	Aufbau einer SWT-Applikation.....	210
7.3.2	SWT-Komponenten.....	211
7.3.3	Layoutmanager.....	226
7.3.4	Ereignisverarbeitung.....	234
7.4	Der Visual Editor für SWT.....	238
7.4.1	Installation und Aufruf.....	238
7.4.2	Erstellen einer grafischen Benutzeroberfläche.....	240
7.4.3	Ereignisverarbeitung.....	243
7.5	Der SWT Designer.....	246
7.5.1	Entwicklung eines GUI mit dem SWT Designer.....	246
7.5.2	Codegenerierung.....	250
7.5.3	Beispiel.....	252
7.6	Der SWT GUI Builder.....	261
7.6.1	Entwicklung eines GUI mit dem SWT GUI Builder.....	262
7.6.2	Codegenerierung.....	265
<b>8</b>	<b>Eclipse und J2EE.....</b>	<b>271</b>
8.1	XML.....	272
8.1.1	Einführung.....	272
8.1.2	Bestandteile eines XML-Dokuments.....	273
8.1.3	Plug-ins für das Bearbeiten von XML-Dokumenten.....	278
8.2	Web-Anwendungen.....	282
8.2.1	Beispiel einer Web-Anwendung mit Servlets.....	283
8.2.2	Beispiel einer Web-Anwendung mit Java Server Pages.....	289
8.2.3	Plug-in zur Entwicklung von Web-Anwendungen.....	291
8.2.4	Beispielanwendung Kreditkartenprüfer.....	299
8.2.5	JSP-Editor.....	304
8.3	J2EE-Anwendungen mit Enterprise Java Beans.....	305
8.3.1	Grundlegende Begriffe.....	305
8.3.2	Enterprise Java Bean (EJB).....	308
8.3.3	Zugriff eines Clients auf eine EJB.....	310
8.3.4	Session Beans.....	312
8.3.5	Plug-in für die Entwicklung und das Deployment von Session Beans.....	319

8.3.6	Client für den Remote-Zugriff auf eine EJB.....	327
8.3.7	Web-Client für den Zugriff auf eine EJB.....	332
8.3.8	Datenbankzugriff aus einer EJB.....	339
8.3.9	Entity Beans.....	346
8.3.10	Entity Bean (Container Managed Persistence).....	348
8.3.11	Entity Bean (BMP).....	362
8.3.12	Java Messaging Service.....	372
8.3.13	Message Driven Beans.....	374
8.3.14	Senden einer Nachricht an eine Message Driven Bean.....	376
<b>9</b>	<b>Web Services.....</b>	<b>383</b>
9.1	Grundlagen.....	383
9.2	Begriffe.....	384
9.2.1	SOAP.....	384
9.2.2	WSDL.....	385
9.2.3	JAX.....	385
9.3	Entwicklung eines Web Services.....	386
9.3.1	Aufbau eines WSDL-Dokuments.....	386
9.3.2	Erzeugen eines Web Services aus einem WSDL-Dokument.....	391
9.3.3	Entwicklung eines Clients für den Web Service.....	397
9.3.4	Erzeugen eines WSDL-Dokuments aus einer Java-Klasse.....	399
	<?	
<b>10</b>	<b>Plug-in-Entwicklung für die Eclipse Workbench.....</b>	<b>405</b>
10.1	In die Tiefen der Oberfläche.....	405
10.2	Everything is a Plug-in.....	405
10.3	Eine Outline View für den Text Editor.....	406
10.3.1	Adapter: Dynamische Erweiterung von Klassen.....	407
10.3.2	Ein eigener Adapter.....	409
10.3.3	Alternative 1: Überschreiben der Methode getAdapter.....	416
10.3.4	Alternative 2: Der Adapter Manager.....	418

10.4	Extensions und Extension Points.....	428
10.4.1	Ein Extension Point wird geboren.....	435
10.4.2	Schritt 1: Definition des Interface für den Extension Point.....	435
10.4.3	Schritt 2: Definition des Extension Point.....	436
10.4.4	Schritt 3: Implementierung des Extension Point.....	440
10.4.5	Eine Extension für den neuen Extension Point.....	442
10.5	Der eigene Editor.....	444
10.5.1	Nichts einfacher als ein eigener Text Editor.....	444
10.5.2	Wir erweitern den Editor.....	450
10.6	Und Action ..: Actions in der Eclipse Workbench (Teil 1).....	457
10.7	Einschub: Aufbau der Eclipse-Oberfläche.....	461
10.8	Und Action ..: Actions in der Eclipse Workbench (Teil 2).....	480
10.8.1	Ein Button mit persistentem Status.....	482
10.9	Page Book View: Ein Buch mit sieben Seiten.....	489
10.9.1	Action Delegates oder „Wie kommt man von außen in das Buch?“.....	491
10.10	Markers und Annotations.....	506
10.10.1	Marker für Properties.....	513
10.11	OSGi.....	532
10.11.1	Bundles.....	533
10.11.2	Services.....	541
10.11.3	Die Benutzung eines Service.....	546
10.12	Fazit und Ausblick.....	550
<sup>a</sup> 11	<b>Plug-In=Entwicklung für das JDT.....</b>	<b>555</b>
11.1	Soap.....	555
11.2	Apache Axis.....	557
11.2.1	Installation von Tomcat.....	557
11.2.2	Installation von Axis.....	559
11.3	Das Web Service Deployment Plug-in.....	561
11.4	Den Rahmen erzeugen.....	561
11.5	Die Funktionalität implementieren.....	567
11.6	Eingabe der Konfigurationsparameter.....	573
11.6.1	Speichern und Laden der Konfigurationsparameter.....	585



## Inhaltsverzeichnis

11.7	Generierung des Deployment Descriptor.....	588
11.8	Registrierung des Web Service bei Apache Axis.....	592
11.9	Generierung eines Ant-Skripts.....	593
11.10	Ausführen des Ant-Skripts.....	595
11.11	Das fertige Plug-in.....	596
11.12	Ein Extension Point für die Generierung von Clients.....	600
11.13	Die Implementierung des Extension Points.....	603
11.14	Ein Generator für SWT Clients.....	605
11.15	Das SWT-Client Generator Plug-in.....	612
11.16	Die Erzeugung von Java-Code über die Java-Element-API.....	613
11.17	So ganz nebenbei.....	618
11.18	Code-Analyse über AST.....	620
11.19	Eine View zur Anzeige des Syntaxbaums.....	621
11.20	Die Erzeugung von Java-Code über AST.....	631
<b>12</b>	<b>Frameworkbasierte GUI-Entwicklung.....</b>	<b>635</b>
12.1	Kapitelüberblick.....	635
12.1.1	Themenüberblick.....	635
12.1.2	Kapitelstruktur.....	635
12.1.3	Ziele des Kapitels.....	635
12.2	Beispiel Personeneditor.....	636
12.2.1	Der Editor - Grundlegendes und Implementierung.....	636
12.2.2	Die Datenklasse.....	637
12.2.3	Editormodell.....	637
12.2.4	Anzeigen des Editors.....	639
12.2.5	Diskussion.....	640
12.3	Allgemeines zur frameworkbasierten GUI-Entwicklung.....	640
12.3.1	Motivation und Hintergrund.....	640
12.3.2	Anforderungen an das Framework.....	641
12.3.3	Technische Implementierungsprinzipien.....	641
12.3.4	Entwicklungsprozess.....	642
12.3.5	Vergleich mit anderen Ansätzen/Frameworks/ Patterns.....	642
12.4	Frameworkfeatures und weitere Beispiele.....	643
12.4.1	Tree Views.....	643

12.4.2 Editoren.....	653
12.5 Warum das alles? Vorteile.....	659
12.5.1 Testen von GUIs.....	659
12.5.2 Berechtigungen.....	660
12.5.3 Events.....	661
12.6 Das Framework und die Eclipse-Philosophie.....	662
12.7 Zusammenfassung.....	663
13.1 Kapitelüberblick.....	665
13.1.1 Themenüberblick.....	665
13.1.2 Kapitelstruktur.....	665
13.1.3 Ziele des Kapitels.....	666
13.2 Anwendungen auf Basis der Plug-in-Technologie.....	666
13.2.1 Motivation.....	666
13.2.2 Sidebar: Bundles versus Plug-ins.....	667
13.2.3 Hello World.....	667
13.2.4 Strukturierung von Anwendungen.....	671
13.2.5 Tipps, Tricks und mögliche Fallen.....	673
13.3 Die Eclipse Rich Client Platform.....	675
13.3.1 Motivation.....	675
13.3.2 Hello World.....	675
13.3.3 Menüs.....	679
13.3.4 Views und Editors.....	683
-13.4 Weitere Themen.....	685
13.4.1 Swing und/oder SWT.....	685
13.4.2 Eclipse im Client-Server-Umfeld.....	686
13.4.3 Eclipse in der Projektpraxis.....	687
13.4.4 Ausblick.....	689
13.5 Zusammenfassung.....	689
14.1 Kapitelüberblick.....	691
14.1.1 Themenüberblick.....	691

## Inhaltsverzeichnis

14.1.2	Kapitelstruktur.....	691
14.2	Die Refactorings im Überblick.....	692
14.3	Refactoring: Konvertiere Standalone-Anwendung in Single-Plug-in-Eclipse-Anwendung.....	692
14.3.1	Motivation.....	693
14.3.2	Mechanics.....	693
14.3.3	Beispiel.....	693
14.4	Refactoring: Extrahiere Bibliotheks-Plug-in.....	697
14.4.1	Motivation.....	697
14.4.2	Mechanics.....	697
14.4.3	Beispiel.....	697
14.5	Refactoring: Extrahiere Erweiterungs-Plug-in.....	698
14.5.1	Motivation.....	699
14.5.2	Mechanics.....	699
14.5.3	Beispiel.....	699
14.6	Refactoring: Verallgemeinere Extension Point.....	702
14.6.1	Motivation.....	702
14.6.2	Mechanics.....	703
14.6.3	Beispiel.....	703
14.7	Refactoring: Konvertiere Anwendung in RCP-Anwendung.....	704
14.7.1	Motivation.....	704
14.7.2	Mechanics.....	704
14.7.3	Beispiel.....	705
14.8	Refactoring: Ersetze Swing-Fenster durch RCP View.....	706
14.8.1	Motivation.....	706
14.8.2	Mechanics.....	707
14.8.3	Beispiel.....	707
14.9	Zusammenfassung.....	708