

Inhaltsverzeichnis

Vorwort	15
Einleitung	17
Konventionen im Buch	19
Onlinebeispiele	20
1 Überblick	21
1.1 Ziele des Buchs	23
1.2 Plug-In	25
1.3 Eclipse kurz und knapp	25
Teil I Zyklus 0: Hello World	29
2 Eclipse für die Plug-In-Entwicklung einrichten	31
2.1 Einen Arbeitsbereich einrichten	31
2.2 Quelldateien durchsuchen	31
2.3 Wegweiser	33
3 Hello World	35
3.1 Deklaration und Implementierung trennen	36
3.2 Hello-Schaltfläche	37
3.3 »Hello« sagen	43
3.4 Wegweiser	50
Teil II Zyklus 1: Basis-Plug-In	51
4 Der Plan	53
4.1 JUnit am Beispiel	53
4.2 JUnit-Integration	55
5 Ein Menübefehl zum Starten von Tests	57
5.1 Zusammenfassung	62
5.2 Wegweiser	62

6	Das Verhalten des Menübefehls implementieren	63
6.1	Zusammenfassung	73
6.2	Wegweiser	73
7	Die Ergebnisse anzeigen	75
7.1	Zusammenfassung	79
7.2	Wegweiser	79
8	Einen Erweiterungspunkt definieren	81
8.1	Zusammenfassung	90
8.2	Wegweiser	90
9	Erweiterungen benachrichtigen	91
9.1	Zusammenfassung	92
10	Veröffentlichen	93
10.1	Das Plug-In verpacken	93
10.2	Das Plug-In in einem Feature verpacken	96
10.3	Erweitern	99
10.4	Wegweiser	101
11	Zyklus 1 schließen	103
12	Zwischenspiel: Test-orientierte Plug-In-Entwicklung	105
12.1	PDE JUnit	106
12.2	Die Einrichtung für ein Testprojekt	111
12.3	Die JUnit-Plug-In-Erweiterung testen	113
12.4	Und jetzt ...	117
Teil III	Zyklus 2: Der Rest des Plug-Ins	119
13	Ergebnisse anzeigen	121
13.1	Eine Ansichtserweiterung	121
13.2	Auf den Testfortschritt hören	125
13.3	Farben ändern	127
13.4	Zusammenfassung	132

14 Menüerweiterungen	133
14.1 Kontextmenüs erzeugen	133
14.2 Menüs mit Befehlen erweitern	138
14.3 Zusammenfassung	141
14.4 Wegweiser	141
15 Gescheiterte Tests sind Compilerfehler	143
16 Testfehler als Markierungen	145
16.1 Test für eine Markierung	145
16.2 Das Projekt übergeben	148
16.3 Markierungen erzeugen	150
16.4 Markierungen löschen	153
16.5 Markierungsbilder	155
16.6 Auflösen von Markierungen	156
16.7 Zusammenfassung	159
17 Tests finden	161
17.1 Zusammenfassung	164
18 Builder und Naturen	165
18.1 Ressourcen-Listener und Builder	165
18.2 Builder mit Naturen konfigurieren	167
18.3 Zusammenfassung	173
18.4 Wegweiser	173
19 Autotest-Eigenschaft	175
19.1 Zusammenfassung	183
19.2 Wegweiser	183
20 Ausnahmebehandlung	185
20.1 IStatus, CoreException	185
20.2 Ausnahmen in einem Fehlerdialogfeld darstellen	187
20.3 Fehler protokollieren	188
20.4 Zusammenfassung	190



21 Tracing – ein Plug-In ausstatten	191
21.1 Zusammenfassung	194
21.2 Wegweiser	194
22 Markierungsauflösung – eine lang laufende Operation aufrufen	195
22.1 Markierungsauflösung testen	195
22.2 Erstellen und neu starten	196
22.3 Fortschritt anzeigen	198
22.4 Den Test erneut ausführen	201
22.5 Zusammenfassung	202
22.6 Wegweiser	202
23 Testberichtsansicht – JFace einsetzen	203
23.1 TestResult	204
23.2 Der Test	205
23.3 Die Ansicht	207
23.4 TestReportLabelProvider	211
23.5 TestReportContentProvider	214
23.6 Ereignisse behandeln	220
23.7 Zusammenfassung	225
23.8 Wegweiser	225
24 Ein einfacher Editor, um Tests auszuschließen	227
24.1 Einen Editor hinzufügen	229
24.2 Einen Erweiterer hinzufügen	233
24.3 Zusammenfassung	238
24.4 Wegweiser	238
25 Noch einmal: ResultView – Änderungen beobachten	241
25.1 Farben testen	241
25.2 Änderungen beobachten	242

25.3	Auf Änderungen reagieren	244
25.4	Zusammenfassung	248
25.5	Wegweiser	248
26	Perspektiven	249
26.1	Ansichten in Perspektiven	249
26.2	Das Menü Show View	251
26.3	Zusammenfassung	254
26.4	Wegweiser	254
27	Hilfe	255
27.1	Hilfe auf oberster Ebene	255
27.2	Integrierte Hilfe	258
27.3	Kontextabhängige Hilfe	259
27.4	Zusammenfassung	263
27.5	Wegweiser	264
28	Internationalisierung und Eingabehilfen	265
28.1	Zeichenfolgen aus Manifest-Dateien externalisieren	265
28.2	Zeichenfolgen aus Code externalisieren	266
28.3	Eingabehilfen	268
28.4	Zusammenfassung	270
28.5	Wegweiser	270
29	Ein Plug-In für andere Programmierer veröffentlichen	271
29.1	Die API definieren	271
29.2	Klassen exportieren	272
29.3	Veröffentlichte und interne Pakete trennen	273
29.4	Kern und Benutzeroberfläche trennen	275
29.5	Einen Erweiterungspunkt veröffentlichen – Erweiterungspunktschemas	276
29.6	Zusammenfassung	280
29.7	Wegweiser	280



30 Zyklus 2 schließen	281
30.1 Erweitern	282
30.2 Das Plug-In erneut installieren	284
30.3 Was kommt als Nächstes?	286
Teil IV Zyklus 3: Mustergeschichten	287
31 Laufzeitkern – IAdaptable	289
31.1 Erweiterungsobjekt/ Erweiterungsschnittstelle	291
31.2 Schnittstellen mit IAdaptable offen legen	293
31.3 Adapterfabriken – Schnittstellen zu vorhandenen Typen hinzufügen	294
32 Arbeitsbereichskern – Ressourcen	299
32.1 Auf Dateisystemressourcen zugreifen – Proxy und Brücke	299
32.2 Der Arbeitsbereich – Kompositum	303
32.3 Den Ressourcenbaum traversieren – Besucher	304
32.4 Ressourcenänderungen verfolgen – Beobachter	306
32.5 Änderungen im Stapel verarbeiten – die Execute Around-Methode	308
33 Java-Kern	311
33.1 Von Ressourcen zu Java-Elementen – Adapter	311
33.2 Java-Elemente – (virtueller) Proxy	313
33.3 Der Java-Elementbaum – Kompositum	316
33.4 Typhierarchie – eine Zuordnung objektivieren	317
33.5 Das Java-Modell traversieren	318
33.6 Java-Elementänderungen verfolgen – Beobachter	318
33.7 Ergebnisse sammeln – Erbauer	320
33.8 AST-Analyse – Besucher	321



34 Standard Widget Toolkit – SWT	327
34.1 Komponenten zusammenstellen – Kompositum	327
34.2 Das Layout definieren – Strategie	331
34.3 Auf Ereignisse reagieren – Beobachter	334
35 JFace – Frameworks der Benutzeroberfläche	337
35.1 Betrachter: Eine Komponente mit einem Modell verbinden – austauschbare Adapter	337
35.2 Betrachter: Anpassen ohne Subclassing – Strategie	342
35.3 Aktionen – Befehl	343
36 Benutzeroberfläche der Workbench	347
36.1 Die Lazy Loading-Regel implementieren – virtuelle Proxies	347
36.2 Den Zustand der Benutzeroberfläche dauerhaft speichern – Memento	349
36.3 Workbench-Dienste – IAdaptable	352
37 Zyklus 3 schließen	357
37.1 Letzte Wegweiser	358
37.2 Eine Einladung zum Erweitern	360
Teil V Anhänge	363
A TestRunner-Details	365
A.1 TestRunner	365
A.2 SocketTestRunner	370
B Die TestProject-Einrichtung	373
C AutoTestBuilder mit Ausschlussunterstützung	377
D Quellen	379
Stichwortverzeichnis	381
Danksagung	403