

# Inhaltsverzeichnis

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

<b>6 Das Verhalten des Menübefehls implementieren</b>	<b>63</b>
6.1 Zusammenfassung	73
6.2 Wegweiser	73
<b>7 Die Ergebnisse anzeigen</b>	<b>75</b>
7.1 Zusammenfassung	79
7.2 Wegweiser	79
<b>8 Einen Erweiterungspunkt definieren</b>	<b>81</b>
8.1 Zusammenfassung	90
8.2 Wegweiser	90
<b>9 Erweiterungen benachrichtigen</b>	<b>91</b>
9.1 Zusammenfassung	92
<b>10 Veröffentlichen</b>	<b>93</b>
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
<b>11 Zyklus 1 schließen</b>	<b>103</b>
<b>12 Zwischenspiel: Test-orientierte Plug-In-Entwicklung</b>	<b>105</b>
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
<b>Teil III Zyklus 2: Der Rest des Plug-Ins</b>	<b>119</b>
<b>13 Ergebnisse anzeigen</b>	<b>121</b>
13.1 Eine Ansichtserweiterung	121
13.2 Auf den Testfortschritt hören	125
13.3 Farben ändern	127
13.4 Zusammenfassung	132

<b>14 Menüerweiterungen</b>	<b>133</b>
14.1 Kontextmenüs erzeugen	133
14.2 Menüs mit Befehlen erweitern	138
14.3 Zusammenfassung	141
14.4 Wegweiser	141
<b>15 Gescheiterte Tests sind Compilerfehler</b>	<b>143</b>
<b>16 Testfehler als Markierungen</b>	<b>145</b>
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
<b>17 Tests finden</b>	<b>161</b>
17.1 Zusammenfassung	164
<b>18 Builder und Naturen</b>	<b>165</b>
18.1 Ressourcen-Listener und Builder	165
18.2 Builder mit Naturen konfigurieren	167
18.3 Zusammenfassung	173
18.4 Wegweiser	173
<b>19 Autotest-Eigenschaft</b>	<b>175</b>
19.1 Zusammenfassung	183
19.2 Wegweiser	183
<b>20 Ausnahmebehandlung</b>	<b>185</b>
20.1 IStatus, CoreException	185
20.2 Ausnahmen in einem Fehlerdialogfeld darstellen	187
20.3 Fehler protokollieren	188
20.4 Zusammenfassung	190



<b>21 Tracing – ein Plug-In ausstatten</b>	<b>191</b>
21.1 Zusammenfassung	194
21.2 Wegweiser	194
<b>22 Markierungsauflösung – eine lang laufende Operation aufrufen</b>	<b>195</b>
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
<b>23 Testberichtsansicht – JFace einsetzen</b>	<b>203</b>
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
<b>24 Ein einfacher Editor, um Tests auszuschließen</b>	<b>227</b>
24.1 Einen Editor hinzufügen	229
24.2 Einen Erweiterer hinzufügen	233
24.3 Zusammenfassung	238
24.4 Wegweiser	238
<b>25 Noch einmal: ResultView – Änderungen beobachten</b>	<b>241</b>
25.1 Farben testen	241
25.2 Änderungen beobachten	242

25.3	Auf Änderungen reagieren	244
25.4	Zusammenfassung	248
25.5	Wegweiser	248
<b>26</b>	<b>Perspektiven</b>	<b>249</b>
26.1	Ansichten in Perspektiven	249
26.2	Das Menü Show View	251
26.3	Zusammenfassung	254
26.4	Wegweiser	254
<b>27</b>	<b>Hilfe</b>	<b>255</b>
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
<b>28</b>	<b>Internationalisierung und Eingabehilfen</b>	<b>265</b>
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
<b>29</b>	<b>Ein Plug-In für andere Programmierer veröffentlichen</b>	<b>271</b>
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



<b>30 Zyklus 2 schließen</b>	<b>281</b>
30.1 Erweitern	282
30.2 Das Plug-In erneut installieren	284
30.3 Was kommt als Nächstes?	286
<b>Teil IV Zyklus 3: Mustergeschichten</b>	<b>287</b>
<b>31 Laufzeitkern – IAdaptable</b>	<b>289</b>
31.1 Erweiterungsobjekt/ Erweiterungsschnittstelle	291
31.2 Schnittstellen mit IAdaptable offen legen	293
31.3 Adapterfabriken – Schnittstellen zu vorhandenen Typen hinzufügen	294
<b>32 Arbeitsbereichskern – Ressourcen</b>	<b>299</b>
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
<b>33 Java-Kern</b>	<b>311</b>
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



<b>34 Standard Widget Toolkit – SWT</b>	<b>327</b>
34.1 Komponenten zusammenstellen – Kompositum	327
34.2 Das Layout definieren – Strategie	331
34.3 Auf Ereignisse reagieren – Beobachter	334
<b>35 JFace – Frameworks der Benutzeroberfläche</b>	<b>337</b>
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
<b>36 Benutzeroberfläche der Workbench</b>	<b>347</b>
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
<b>37 Zyklus 3 schließen</b>	<b>357</b>
37.1 Letzte Wegweiser	358
37.2 Eine Einladung zum Erweitern	360
<b>Teil V Anhänge</b>	<b>363</b>
<b>A TestRunner-Details</b>	<b>365</b>
A.1 TestRunner	365
A.2 SocketTestRunner	370
<b>B Die TestProject-Einrichtung</b>	<b>373</b>
<b>C AutoTestBuilder mit Ausschlussunterstützung</b>	<b>377</b>
<b>D Quellen</b>	<b>379</b>
<b>Stichwortverzeichnis</b>	<b>381</b>
Danksagung	403