

Inhalt

	Einleitung	11
	Die Idee des Buchs.....	11
	Zielgruppe des Buchs.....	11
	Aufbau des Buchs.....	12
	Danksagung.....	13
1	Vorbereitungen	15
1.1	Was ist Eclipse?	17
	1.1.1 Die Vorgeschichte.....	17
	1.1.2 Die Lizenz.....	17
	1.1.3 Die Produktqualität.....	18
	1.1.4 Die Organisation.....	20
1.2	Wie funktioniert Eclipse?	21
	1.2.1 Die Architektur.....	21
	1.2.2 Die Plug-ins.....	23
	1.2.3 Was fehlt.....	24
1.3	Installation und Konfiguration	24
	1.3.1 Voraussetzungen.....	24
	1.3.2 Download.....	25
	1.3.3 Installation.....	27
	1.3.4 Migration.....	28
	1.3.5 Konfiguration.....	29
	1.3.6 Aufrufoptionen.....	30
	1.3.7 Mehrfachaufrufe.....	31
2	Die Benutzeroberfläche	33
2.1	Das Grundkonzept	35
	2.1.1 Die Workbench.....	35
	2.1.2 Perspektiven.....	37
	2.1.3 Sichten.....	45
2.2	Editoren	49
	2.2.1 Dateitypen.....	50
	2.2.2 Dateien.....	53
	2.2.3 Darstellung und Sichtbarkeit.....	57
	2.2.4 Editornavigation.....	58

2.3	Suchen und Ersetzen.....	60
2.3.1	Find/Replace.....	60
2.3.2	Search.....	62
2.4	Hilfen.....	65
2.4.1	Onlinehilfe.....	65
2.4.2	Starthilfe.....	67
2.4.3	Haupthilfe.....	68
2.4.4	Spezialhilfen.....	73
2.4.5	Softwareupdate.....	74
2.4.6	Hilfekonfiguration.....	76
3	Ein Java-Projekt	81
3.1	DerWorkspace.....	83
3.1.1	Aufgaben.....	83
3.1.2	Einsatz und Sicherung.....	85
3.1.3	Projektverwaltung.....	86
3.2	Programmerstellung.....	92
3.2.1	Die Java-Perspektive.....	92
3.2.2	Die Quelldateien.....	97
3.2.3	Der Editor.....	100
3.2.4	Der Quellcode.....	108
3.3	Programmausführung.....	118
3.3.1	Programmstart.....	118
3.3.2	Debugging.....	121
4	Java-Entwicklungswerkzeuge	129
4.1	Das Beispielprojekt.....	131
4.1.1	Die Projektaufgabe.....	131
4.1.2	Die Projektstruktur.....	132
4.1.3	Der Programmcode.....	132
4.2	Projektanlage.....	136
4.2.1	Übernahme der Quelldateien.....	136
4.2.2	Startkonfiguration des Projekts.....	141
4.3	Wer sucht.....	144
4.3.1	Filter.....	144
4.3.2	Wo finde ich ...?.....	149
4.3.3	Die Java-Suche.....	153
4.3.4	Die Navigation.....	155
4.3.5	Eine Gesamtschau.....	159
4.4	Codebearbeitung.....	160
4.4.1	Erweiterung geerbter Methoden.....	160
4.4.2	Getter- und Setter-Methoden.....	162
4.4.3	Mehr über Konstruktoren.....	164

4.4.4	Externalisierung.....	167
4.4.5	Schadensbegrenzung.....	170
4.5	Code-Templates.....	175
4.5.1	Nutzung vorhandener Templates.....	175
4.5.2	Verwaltung der Templates.....	176
4.5.3	Anlage eines Templates.....	177

5 Refactoring 179

5.1	Die Idee des Refactorings.....	181
5.1.1	Die Idee des Refactorings.....	181
5.1.2	Unser Refactoring-Projekt.....	183
5.2	Einen Schritt zurück.....	186
5.2.1	Edit • Undo oder Refactor • Undo?.....	186
5.2.2	Vergleiche.....	187
5.2.3	Projektkopien.....	188
5.3	Factory-Methoden und Umbenennungen.....	191
5.3.1	Factory-Methoden.....	191
5.3.2	Umbenennungen.....	193
5.4	Die Ein- und Ausgabe-Interfaces.....	197
5.4.1	Extraktion von Interfaces.....	197
5.4.2	Erweiterung der Interfaces.....	199
5.5	Änderungen an Lottozahlen.....	200
5.5.1	Eine neue Signatur.....	200
5.5.2	Anlage von Exceptions.....	204
5.6	Abschlussarbeiten.....	208
5.6.1	Den Code umstrukturieren und erweitern.....	208
5.6.2	Neue Methoden extrahieren.....	209
5.7	Pakete und Dateien.....	214
5.7.1	Die neue Struktur.....	214
5.7.2	Anlage der Namensräume.....	215
5.7.3	Umorganisation der Dateien.....	215
5.8	Weitere Werkzeuge.....	218
5.8.1	Inline-Code anlegen.....	218
5.8.2	Refactoring von Feldern.....	219
5.8.3	Umwandlung innerer und anonymer Klassen.....	222
5.8.4	Änderung in Klassenhierarchien.....	224

Fehlersuche 231

6.1	Interaktives Debugging.....	233
6.1.1	Vorgehensweisen.....	233
6.1.2	Die Debug-Sitzung.....	234
6.1.3	Die Debug-Perspektive.....	237

6.2	Debuggingtechniken.....	242
6.2.1	Breakpoints.....	242
6.2.2	Variablenüberwachung.....	249
6.2.3	Ausdrucksüberwachung.....	253
6.2.4	Veränderungen an Variablen und Programmcode.....	257
6.3	Eine Beispielsitzung.....	258
6.3.1	Breakpoints.....	259
6.3.2	In compareNumbersO.....	260
6.3.3	In AusgabeO.....	261
6.3.4	Einkreisen der Ursache.....	261
6.4	Anwendungstest mit JUnit.....	265
6.4.1	Was ist JUnit?.....	266
6.4.2	Der Aufbau von JUnit.....	267
6.4.3	Der Einsatz von JUnit.....	268
6.4.4	Test von Lottobude.....	275
6.4.5	Anlage einer Testsuite.....	279

7 Übersetzungssteuerung 281

7.1	Übersetzungseinstellungen.....	283
7.1.1	Der Run-Dialog.....	283
7.1.2	Definition einer Startkonfiguration.....	284
7.2	Ant und Eclipse.....	298
7.2.1	Was ist Ant?.....	298
7.2.2	Die Ant-Werkzeuge.....	302
7.2.3	Der Build-Lauf.....	307
7.2.4	Der Build-Dialog.....	310
7.2.5	Ant als Project Builder.....	314
7.3	Ant im Einsatz.....	315
7.3.1	Die Aufgabe.....	315
7.3.2	Anlage der Steuerdatei.....	316
7.3.3	Der Build-Lauf.....	319

8 Plug-ins und GUI-Tools 321

8.1	Plug-ins.....	323
8.1.1	Grundlagen.....	323
8.1.2	Der Plug-In-Wizard.....	324
8.2	DerVisual Editor.....	331
8.2.1	Quellen.....	331
8.2.2	Installation.....	332
8.2.3	Erster Test.....	332
8.3	Einsatz des Editors.....	335
8.3.1	DerVisual Editor.....	335
8.3.2	Java-Beans-Sicht.....	344

8.4	Ein GUI für Lottobude.....	347
8.4.1	Die Aufgabenstellung.....	347
8.4.2	Das Grundgerüst.....	347
8.4.3	Der Programmcode.....	350

Teamarbeit 359

9.1	Verbindungsaufnahme.....	361
9.1.1	Die Verbindungsaufnahme.....	361
9.1.2	Verwaltung der Verbindung.....	363
9.2	Projekte bereitstellen.....	364
9.2.1	Projekte einpflegen.....	364
9.2.2	Projektinformationen.....	367
9.2.3	Die Repository-Sicht.....	368
9.2.4	Die Menüs.....	369
9.3	Code-Änderungen.....	369
9.3.1	Dateiänderungen pflegen.....	369
9.3.2	Änderungen vergleichen und zurücknehmen.....	372
9.3.3	Änderungen zurücknehmen.....	375
9.3.4	Patches.....	376
9.4	Markierungen.....	379
9.4.1	Versionstags.....	379
9.4.2	Zeitmarken.....	380
9.4.3	Entwicklungszweige.....	381
9.5	Das Repository im Überblick.....	382
9.5.1	Die CVS-Repository-Exploring-Perspektive.....	383
9.5.2	Annotations.....	383
9.5.3	CVS Resource History.....	384
9.5.4	CVS-Repositories.....	385
9.5.5	Die Check-out-Kommandos.....	386
9.5.6	Die Tag-Kommandos.....	388
9.5.7	Die Compare-Kommandos.....	388
9.5.8	Die Branch-Kommandos.....	390

Index 391