

HANSER

Dierk König, Andrew Glover, Paul King, Guillaume Laforge, Jon
Skeet

Groovy im Einsatz

ISBN-10: 3-446-41238-7
ISBN-13: 978-3-446-41238-5

Inhaltsverzeichnis

Weitere Informationen oder Bestellungen unter
<http://www.hanser.de/978-3-446-41238-5>
sowie im Buchhandel



Inhalt

- Vorwort XV**
- Vorwort zur deutschen Ausgabe..... XVII**
- Einleitung XIX**
- Danksagungen XXIII**
- Über dieses BuchXXV**
- Die AutorenXXIX**

- 1 Ihr Weg zu Groovy..... 1**
- 1.1 Die Groovy-Story.....2
 - 1.1.1 Was ist Groovy?.....3
 - 1.1.2 Gutes Zusammenspiel mit Java: nahtlose Integration.....4
 - 1.1.3 Power in Ihrem Code: eine reichhaltige Sprache5
 - 1.1.4 Community-gesteuert, aber von Unternehmen gestützt.....8
- 1.2 Was Groovy für Sie tun kann.....9
 - 1.2.1 Groovy für Java-Profis.....9
 - 1.2.2 Groovy für Skriptprogrammierer10
 - 1.2.3 Groovy für pragmatische, extreme und agile Programmierer.....11
- 1.3 Groovy ausführen.....12
 - 1.3.1 „Hello World“ auf groovysh12
 - 1.3.2 Die groovyConsole.....15
 - 1.3.3 Der Befehl groovy17
- 1.4 Groovy kompilieren und ausführen18
 - 1.4.1 Groovy mit groovyc kompilieren.....18
 - 1.4.2 Ein kompiliertes Groovy-Skript mit Java ausführen.....19
 - 1.4.3 Kompilieren und Ausführen mit Ant19
- 1.5 IDE- und Editor-Unterstützung für Groovy21
 - 1.5.1 Das Plugin für IntelliJ IDEA21

| | | |
|--|--|-----------|
| 1.5.2 | Das Eclipse-Plugin..... | 22 |
| 1.5.3 | Groovy-Unterstützung in anderen Editoren..... | 22 |
| 1.6 | Zusammenfassung | 23 |
| Teil 1: Die Programmiersprache Groovy | | 25 |
| 2 | Die Ouvertüre: Groovy-Grundlagen | 27 |
| 2.1 | Allgemeines Aussehen des Codes | 28 |
| 2.1.1 | Kommentare in Groovy | 28 |
| 2.1.2 | Vergleich der Syntax von Groovy und Java | 29 |
| 2.1.3 | Kurz und gut | 29 |
| 2.2 | Testen der Sprache durch Zusicherungen..... | 30 |
| 2.3 | Groovy auf einen Blick..... | 33 |
| 2.3.1 | Klassen deklarieren..... | 33 |
| 2.3.2 | Verwendung von Skripten | 34 |
| 2.3.3 | GroovyBeans | 35 |
| 2.3.4 | Umgang mit Text | 36 |
| 2.3.5 | Zahlen sind Objekte | 37 |
| 2.3.6 | Listen, Maps und Ranges..... | 37 |
| 2.3.7 | Code als Objekt: Closures..... | 40 |
| 2.3.8 | Groovy-Kontrollstrukturen | 42 |
| 2.4 | Groovys Platz in der Java-Umgebung | 44 |
| 2.4.1 | Meine Klasse ist deine Klasse | 44 |
| 2.4.2 | GDK: Die Groovy-Bibliothek | 44 |
| 2.4.3 | Der Groovy-Lebenszyklus | 46 |
| 2.5 | Zusammenfassung | 49 |
| 3 | Die einfachen Groovy-Datentypen | 51 |
| 3.1 | Objekte, Objekte überall..... | 51 |
| 3.1.1 | Das Typsystem von Java: Basistypen und Referenztypen..... | 52 |
| 3.1.2 | Die Antwort ist Groovy – alles ist ein Objekt | 52 |
| 3.1.3 | Zusammenwirken mit Java – automatisches Boxing und Unboxing..... | 54 |
| 3.1.4 | Kein zwischengeschaltetes Unboxing | 55 |
| 3.2 | Das Konzept der optionalen Typisierung | 56 |
| 3.2.1 | Typen zuweisen | 56 |
| 3.2.2 | Statische versus dynamische Typisierung | 56 |
| 3.3 | Operatoren überschreiben | 58 |
| 3.3.1 | Überschreibbare Operatoren | 58 |
| 3.3.2 | Überschriebene Operatoren in Aktion | 60 |
| 3.3.3 | Automatische Typumwandlung | 61 |
| 3.4 | Umgang mit Strings | 63 |
| 3.4.1 | Varianten von Stringliteralen..... | 63 |
| 3.4.2 | Der Umgang mit GStrings | 66 |
| 3.4.3 | Von Java zu Groovy | 67 |
| 3.5 | Arbeiten mit regulären Ausdrücken..... | 70 |
| 3.5.1 | Angabe von Patterns in Stringliteralen | 71 |

| | | |
|----------|---|------------|
| 3.5.2 | Anwendung von Patterns | 73 |
| 3.5.3 | Patterns in Aktion..... | 75 |
| 3.5.4 | Patterns und Performance..... | 77 |
| 3.5.5 | Patterns für die Klassifizierung | 78 |
| 3.6 | Arbeiten mit Zahlen | 79 |
| 3.6.1 | Automatische Typumwandlung mit numerischen Operatoren..... | 79 |
| 3.6.2 | GDK-Methoden für numerische Typen | 81 |
| 3.7 | Zusammenfassung | 82 |
| 4 | Die mengenwertigen Groovy-Datentypen | 85 |
| 4.1 | Der Umgang mit Ranges | 86 |
| 4.1.1 | Ranges spezifizieren..... | 87 |
| 4.1.2 | Ranges sind Objekte..... | 88 |
| 4.1.3 | Ranges in Aktion..... | 89 |
| 4.2 | Der Umgang mit Listen..... | 91 |
| 4.2.1 | Listen angeben..... | 91 |
| 4.2.2 | Der Umgang mit Listenoperatoren..... | 92 |
| 4.2.3 | Verwenden von Listenmethoden..... | 95 |
| 4.2.4 | Listen in Aktion..... | 99 |
| 4.3 | Der Umgang mit Maps..... | 101 |
| 4.3.1 | Maps angeben..... | 101 |
| 4.3.2 | Verwendung von Map-Operatoren..... | 103 |
| 4.3.3 | Maps in Aktion..... | 107 |
| 4.4 | Hinweise zu Groovy-Collections | 108 |
| 4.4.1 | Nebenläufige Änderungen..... | 109 |
| 4.4.2 | Der Unterschied zwischen Kopier- und Änderungs-Semantik | 109 |
| 4.5 | Zusammenfassung | 110 |
| 5 | Closures | 111 |
| 5.1 | Eine sanfte Einführung in Closures..... | 112 |
| 5.2 | Ein Fall für Closures | 113 |
| 5.2.1 | Verwendung von Iteratoren..... | 113 |
| 5.2.2 | Ressourcenbehandlung..... | 114 |
| 5.3 | Deklarieren von Closures | 117 |
| 5.3.1 | Die einfache Deklaration..... | 117 |
| 5.3.2 | Zuweisungen zur Deklaration verwenden..... | 118 |
| 5.3.3 | Methoden als Closures referenzieren | 119 |
| 5.3.4 | Die verschiedenen Möglichkeiten im Vergleich..... | 121 |
| 5.4 | Verwendung von Closures | 122 |
| 5.4.1 | Aufrufen einer Closure..... | 122 |
| 5.4.2 | Weitere Closure-Methoden | 124 |
| 5.5 | Gültigkeitsbereiche | 127 |
| 5.5.1 | Der einfache Gültigkeitsbereich einer Variablen..... | 128 |
| 5.5.2 | Der allgemeine Gültigkeitsbereich von Closures..... | 129 |
| 5.5.3 | Gültigkeitsbereiche: der klassische Akkumulator-Test..... | 132 |
| 5.6 | Rückgabe aus Closures | 133 |

| | | |
|----------|--|------------|
| 5.7 | Unterstützung für Entwurfsmuster | 134 |
| 5.7.1 | Beziehung zum Besuchermuster | 135 |
| 5.7.2 | Beziehung zum Erbauermuster | 136 |
| 5.7.3 | Beziehung zu anderen Mustern | 136 |
| 5.8 | Zusammenfassung | 137 |
| 6 | Groovy-Kontrollstrukturen | 139 |
| 6.1 | Die Groovy-Wahrheit | 139 |
| 6.1.1 | Boolesche Tests auswerten | 140 |
| 6.1.2 | Zuweisungen in Booleschen Tests | 141 |
| 6.2 | Bedingte Ausführung | 143 |
| 6.2.1 | Die ganz bescheidene if-Anweisung | 143 |
| 6.2.2 | Der Bedingungsoperator ?: | 144 |
| 6.2.3 | Die switch-Anweisung | 144 |
| 6.2.4 | Plausibilitätsprüfung mit Zusicherungen | 147 |
| 6.3 | Schleifen | 151 |
| 6.3.1 | while-Schleifen | 151 |
| 6.3.2 | for-Schleifen | 151 |
| 6.4 | Blöcke und Methoden abbrechen | 154 |
| 6.4.1 | Normale Beendigung: return/break/continue | 154 |
| 6.4.2 | Exceptions: throw/try-catch-finally | 155 |
| 6.5 | Zusammenfassung | 156 |
| 7 | Dynamische Objektorientierung nach Groovy-Art | 157 |
| 7.1 | Definieren von Klassen und Skripten | 158 |
| 7.1.1 | Definieren von Feldern und lokalen Variablen | 158 |
| 7.1.2 | Methoden und Parameter | 161 |
| 7.1.3 | Mit dem Operator ?. sicher dereferenzieren | 165 |
| 7.1.4 | Konstruktoren | 166 |
| 7.2 | Organisieren von Klassen und Skripten | 169 |
| 7.2.1 | Beziehung zwischen Datei und Klasse | 169 |
| 7.2.2 | Organisieren von Klassen in Packages | 170 |
| 7.2.3 | Weitere Überlegungen zum Klassenpfad | 174 |
| 7.3 | Fortgeschrittene OO-Merkmale | 175 |
| 7.3.1 | Vererbung | 175 |
| 7.3.2 | Interfaces | 176 |
| 7.3.3 | Multimethoden | 177 |
| 7.4 | GroovyBeans | 179 |
| 7.4.1 | Deklaration von Beans | 179 |
| 7.4.2 | Umgang mit Beans | 181 |
| 7.4.3 | Bean-Methoden für alle Objekte | 184 |
| 7.4.4 | Felder, Zugriffsmethoden, Maps und Expando | 185 |
| 7.5 | Power-Features | 186 |
| 7.5.1 | Objekte mit GPaths abfragen | 187 |
| 7.5.2 | Den Spread-Operator injizieren | 191 |
| 7.5.3 | Kategorienmix mit use | 191 |

| | | |
|---|--|------------|
| 7.6 | Meta-Programmierung in Groovy | 194 |
| 7.6.1 | Das Konzept der MetaClass | 194 |
| 7.6.2 | Methoden aufrufen und abfangen | 196 |
| 7.6.3 | Abfangen von Methoden | 198 |
| 7.7 | Zusammenfassung | 201 |
| Teil 2: Die Groovy-Bibliothek..... | | 203 |
| 8 | Builder | 205 |
| 8.1 | Learning by doing – Verwendung eines Builders | 206 |
| 8.2 | Erstellen von Objektbäumen mit NodeBuilder | 209 |
| 8.2.1 | NodeBuilder in Aktion – der Builder-Code | 210 |
| 8.2.2 | Das Builder-Konzept..... | 211 |
| 8.2.3 | Intelligente Builder mit Logik..... | 212 |
| 8.3 | Arbeiten mit MarkupBuilder | 214 |
| 8.3.1 | Erstellen von XML..... | 214 |
| 8.3.2 | Erstellen von HTML | 215 |
| 8.4 | Aufgaben automatisieren mit AntBuilder | 217 |
| 8.4.1 | Von Ant-Skripten zu Groovy-Skripten | 218 |
| 8.4.2 | Wie AntBuilder funktioniert | 219 |
| 8.4.3 | Intelligente Automatisierungsskripte mit Logik | 220 |
| 8.5 | GUIs ganz easy mit SwingBuilder | 221 |
| 8.5.1 | Passwort lesen mit dem SwingBuilder | 222 |
| 8.5.2 | Erstellen von Swing-Widgets..... | 223 |
| 8.5.3 | Widgets anordnen..... | 227 |
| 8.5.4 | Referenzieren von Widgets | 230 |
| 8.5.5 | Swing-Actions..... | 233 |
| 8.5.6 | Verwendung von Modellen | 235 |
| 8.5.7 | Von den Teilen zum Ganzen..... | 237 |
| 8.6 | Eigene Builder erstellen | 243 |
| 8.6.1 | Unterklassen für BuilderSupport..... | 244 |
| 8.6.2 | Das DebugBuilder-Beispiel..... | 245 |
| 8.7 | Zusammenfassung | 247 |
| 9 | Das GDK..... | 249 |
| 9.1 | Objekte | 250 |
| 9.1.1 | Interaktive Objekte..... | 250 |
| 9.1.2 | Bequemlichkeitsmethoden von Object..... | 256 |
| 9.1.3 | Iterative Object-Methoden | 258 |
| 9.2 | Arbeiten mit Dateien und I/O..... | 261 |
| 9.2.1 | Durchquerung des Dateisystems | 263 |
| 9.2.2 | Eingabequellen lesen..... | 265 |
| 9.2.3 | Ausgaben schreiben..... | 266 |
| 9.2.4 | Filter und Konvertierungen | 268 |
| 9.2.5 | Serialisierte Objekt-Streams..... | 269 |
| 9.3 | Threads und Prozesse..... | 270 |

| | | |
|-----------|---|------------|
| 9.3.1 | Groovy-Multithreading | 271 |
| 9.3.2 | Integrieren externer Prozesse | 273 |
| 9.4 | Die Arbeit mit Templates | 277 |
| 9.4.1 | Das Format von Templates | 277 |
| 9.4.2 | Templates in Aktion | 278 |
| 9.4.3 | Templates für Fortgeschrittene | 280 |
| 9.5 | Groovlets | 281 |
| 9.5.1 | „Hello World“ | 282 |
| 9.5.2 | Die Groovlet-Bindung | 284 |
| 9.5.3 | Templates und Groovlets | 287 |
| 9.6 | Zusammenfassung | 289 |
| 10 | Datenbankprogrammierung mit Groovy | 291 |
| 10.1 | Grundlegende Datenbankoperationen | 292 |
| 10.1.1 | Datenbankzugriff einrichten | 293 |
| 10.1.2 | SQL ausführen | 296 |
| 10.1.3 | Daten abfragen | 301 |
| 10.1.4 | Von den Teilen zum Ganzen | 304 |
| 10.2 | DataSets für SQL ohne SQL | 306 |
| 10.2.1 | DataSet-Operationen verwenden | 307 |
| 10.2.2 | DataSets auf Datenbank-Views | 310 |
| 10.3 | Organisieren der Datenbankarbeit | 312 |
| 10.3.1 | Überblick über die Architektur | 312 |
| 10.3.2 | Verhalten der Anwendung spezifizieren | 314 |
| 10.3.3 | Implementierung der Infrastruktur | 315 |
| 10.3.4 | Ein transparentes Domain-Modell | 319 |
| 10.3.5 | Implementierung der Anwendungsschicht | 320 |
| 10.4 | Groovy und ORM | 321 |
| 10.5 | Zusammenfassung | 322 |
| 11 | Groovy-Integration | 325 |
| 11.1 | Vorbereitung auf die Integration | 326 |
| 11.1.1 | Richtig integrieren | 326 |
| 11.1.2 | Abhängigkeiten einrichten | 328 |
| 11.2 | Ausdrücke und Skripte auswerten mit GroovyShell | 328 |
| 11.2.1 | Aller Anfang ist leicht | 329 |
| 11.2.2 | Übergabe von Parametern in einer Bindung | 331 |
| 11.2.3 | Dynamische Klassen zur Laufzeit generieren | 333 |
| 11.2.4 | Skripte parsen | 333 |
| 11.2.5 | Skripte oder Klassen ausführen | 334 |
| 11.2.6 | Weitere Parametrisierung von GroovyShell | 335 |
| 11.3 | Die Groovy-Skript-Engine | 339 |
| 11.3.1 | Einrichten der Engine | 339 |
| 11.3.2 | Skripte ausführen | 340 |
| 11.3.3 | Abweichenden Ressourcenkonnektor definieren | 340 |
| 11.4 | Der GroovyClassLoader | 341 |

| | | |
|----------------|---|------------|
| 11.4.1 | Groovy-Klassen parsen und laden..... | 341 |
| 11.4.2 | Was war zuerst da: Die Henne oder das Ei? | 343 |
| 11.4.3 | Ein benutzerdefinierter Resource-Loader | 346 |
| 11.4.4 | Sicherheit in der Sandbox | 347 |
| 11.5 | Spring-Integration | 350 |
| 11.5.1 | Wiring GroovyBeans..... | 351 |
| 11.5.2 | Aktualisierbare Beans | 353 |
| 11.5.3 | Inline-Skripte..... | 353 |
| 11.6 | Mustang und JSR-223 | 354 |
| 11.6.1 | Einführung in JSR-223..... | 355 |
| 11.6.2 | Der Skript-Engine-Manager und seine Skript-Engines..... | 356 |
| 11.6.3 | Kompilierbare und aufrufbare Skript-Engines..... | 357 |
| 11.7 | Die Wahl des richtigen Integrationsmechanismus..... | 359 |
| 11.8 | Zusammenfassung..... | 360 |
| 12 | Arbeiten mit XML..... | 361 |
| 12.1 | XML-Dokumente lesen..... | 362 |
| 12.1.1 | Arbeiten mit einem DOM-Parser | 363 |
| 12.1.2 | Daten lesen mit dem Groovy-Parser | 367 |
| 12.1.3 | Lesen mit einem SAX-Parser..... | 372 |
| 12.1.4 | Lesen mit einem StAX-Parser..... | 373 |
| 12.2 | XML verarbeiten..... | 375 |
| 12.2.1 | Vor Ort-Verarbeitung..... | 376 |
| 12.2.2 | Streaming-Verarbeitung..... | 378 |
| 12.2.3 | Kombination mit XPath | 383 |
| 12.3 | Verteilte Verarbeitung mit XML | 390 |
| 12.3.1 | Überblick über Webdienste..... | 390 |
| 12.3.2 | RSS und ATOM lesen..... | 391 |
| 12.3.3 | Ein REST-basiertes API..... | 393 |
| 12.3.4 | XML-RPC..... | 396 |
| 12.3.5 | Anwendung von SOAP..... | 399 |
| 12.4 | Zusammenfassung..... | 403 |
| Teil 3: | Groovy für jeden Tag | 405 |
| 13 | Tipps und Tricks..... | 407 |
| 13.1 | Was Sie sich merken sollten | 408 |
| 13.1.1 | Gleichheit versus Identität..... | 408 |
| 13.1.2 | Klammern richtig setzen | 408 |
| 13.1.3 | Rückkehr aus Methoden und Closures..... | 409 |
| 13.1.4 | Methoden aufrufen im Builder-Code | 410 |
| 13.1.5 | Qualifizierter Zugriff auf „this“ | 412 |
| 13.1.6 | Numerische Typen | 413 |
| 13.1.7 | Ant..... | 414 |
| 13.1.8 | Skripte sind Klassen, nur anders | 416 |
| 13.2 | Nützliche Codefragmente..... | 419 |

| | | |
|-----------|--|------------|
| 13.2.1 | Eine Collection mischen | 419 |
| 13.2.2 | Text durcheinanderwürfeln mit regulären Ausdrücken | 420 |
| 13.2.3 | Fortschrittsanzeige auf der Konsole | 420 |
| 13.2.4 | Selbstkommentierende Einzelschritte | 421 |
| 13.2.5 | GString für Fortgeschrittene | 422 |
| 13.3 | Groovy auf der Kommandozeile | 423 |
| 13.3.1 | Kommandozeilenskripte auswerten | 424 |
| 13.3.2 | Die Optionen print und line | 425 |
| 13.3.3 | Der Lauschmodus | 426 |
| 13.3.4 | Vor-Ort-Bearbeitung auf der Kommandozeile | 427 |
| 13.4 | Automatisierungsskripte | 427 |
| 13.4.1 | Konsistente Unterstützung für Kommandozeilenoptionen | 428 |
| 13.4.2 | Klassenpfad mit RootLoader erweitern | 431 |
| 13.4.3 | Ausführungsfahrplan für Skripte | 433 |
| 13.5 | Beispiele für die Automatisierung von Tasks | 435 |
| 13.5.1 | Scraping von HTML-Seiten | 435 |
| 13.5.2 | Automatisieren von Webaktionen | 437 |
| 13.5.3 | Versionskontrolle inspizieren | 439 |
| 13.5.4 | Pragmatische Codeanalyse | 440 |
| 13.5.5 | Weitere interessante Punkte | 441 |
| 13.6 | Layout des Arbeitsbereichs | 443 |
| 13.6.1 | IDE-Setup | 443 |
| 13.6.2 | Debugging | 444 |
| 13.6.3 | Profiling | 448 |
| 13.6.4 | Refactoring | 450 |
| 13.7 | Zusammenfassung | 450 |
| 14 | Unit Testing mit Groovy | 453 |
| 14.1 | Erste Schritte | 454 |
| 14.1.1 | Tests schreiben ist einfach | 454 |
| 14.1.2 | GroovyTestCase: Eine Einführung | 455 |
| 14.1.3 | GroovyTestCase | 456 |
| 14.2 | Unit Testing für Groovy-Code | 457 |
| 14.3 | Unit Testing für Java-Code | 461 |
| 14.4 | Tests organisieren | 464 |
| 14.5 | Fortgeschrittene Testmethoden | 465 |
| 14.5.1 | Testen wird groovy | 466 |
| 14.5.2 | Stubs und Mocks | 467 |
| 14.5.3 | GroovyLogTestCase | 472 |
| 14.6 | IDE-Integration | 474 |
| 14.6.1 | Verwenden von GroovyTestSuite | 474 |
| 14.6.2 | Verwenden von AllTestSuite | 476 |
| 14.6.3 | Fortgeschrittene IDE-Integration | 477 |
| 14.7 | Tools für Groovy-Testing | 479 |
| 14.7.1 | Code-Abdeckung mit Groovy | 479 |
| 14.7.2 | JUnit-Erweiterungen | 482 |

| | | |
|-----------|--|------------|
| 14.8 | Build-Automatisierung..... | 485 |
| 14.8.1 | Build-Integration mit Ant..... | 485 |
| 14.8.2 | Build-Integration mit Maven..... | 486 |
| 14.9 | Zusammenfassung..... | 489 |
| 15 | Windows wird Groovy..... | 491 |
| 15.1 | Scriptom herunterladen und installieren..... | 492 |
| 15.2 | Scriptom intern..... | 493 |
| 15.2.1 | Jacob..... | 493 |
| 15.2.2 | Instanzieren einer ActiveX-Komponente..... | 495 |
| 15.2.3 | Methoden aufrufen..... | 497 |
| 15.2.4 | Zugriff auf Properties und Rückgabewerte..... | 499 |
| 15.2.5 | Event-Unterstützung..... | 500 |
| 15.3 | Praxis-Szenario: Automatisieren der Lokalisierung..... | 502 |
| 15.3.1 | Entwurf des Dokumentformats..... | 503 |
| 15.3.2 | Entwurf des Thesaurus..... | 504 |
| 15.3.3 | Erstellen eines Word-Dokuments..... | 506 |
| 15.3.4 | Erstellung des endgültigen Dokuments..... | 507 |
| 15.4 | Automatisierung weiterer Anwendungen..... | 509 |
| 15.4.1 | Zugriff auf die Windows-Registrierung..... | 509 |
| 15.4.2 | Ein eigenes Automatisierungssystem programmieren..... | 511 |
| 15.5 | Verfügbare Dokumentation..... | 512 |
| 15.6 | Zusammenfassung..... | 513 |
| 16 | Ich sehe das Licht: Grails..... | 515 |
| 16.1 | Die Ausgangssituation..... | 515 |
| 16.1.1 | Grails installieren..... | 516 |
| 16.1.2 | Die ersten Schritte..... | 517 |
| 16.2 | Erstellen des Domain-Modells..... | 519 |
| 16.2.1 | Die Use Cases..... | 520 |
| 16.2.2 | Beziehungen designen..... | 520 |
| 16.3 | Implementierung des Domain-Modells..... | 522 |
| 16.3.1 | Scaffolding der Domain-Klassen..... | 522 |
| 16.3.2 | Scaffolding der Views und Controller..... | 523 |
| 16.3.3 | Testen der Webanwendung..... | 524 |
| 16.3.4 | Vervollständigen des Domain-Modells..... | 526 |
| 16.4 | Anpassen der Views..... | 528 |
| 16.4.1 | Bootstrapping von Daten..... | 528 |
| 16.4.2 | Arbeiten mit Groovy Server Pages..... | 529 |
| 16.4.3 | Arbeiten mit Tag-Bibliotheken..... | 532 |
| 16.5 | Arbeiten mit Controllern und Finder-Methoden..... | 534 |
| 16.6 | Ausarbeitung des Modells..... | 536 |
| 16.7 | Arbeiten mit der Session..... | 538 |
| 16.8 | Das Finish..... | 542 |
| 16.8.1 | Constraints validieren..... | 542 |
| 16.8.2 | Die Anwendung herausgeben..... | 543 |

| | | |
|----------|--|------------|
| 16.8.3 | Abschied | 545 |
| A | Installation und Dokumentation | 549 |
| A.1 | Installation | 549 |
| A.2 | Aktuelle Dokumentation beschaffen | 550 |
| A.2.1 | Online-Ressourcen | 550 |
| A.2.2 | Abonnieren von Mailinglisten | 551 |
| A.2.3 | Verbindung mit dem Forum und Chat | 551 |
| B | Infos zur Programmiersprache Groovy | 553 |
| B.1 | Schlüsselwörter | 554 |
| C | GDK-API-Kurzreferenz | 555 |
| C.1 | Arrays und Primitives | 555 |
| C.2 | Das Package java.lang | 556 |
| C.3 | Das Package java.math | 560 |
| C.4 | Die Packages java.util und java.sql | 561 |
| C.5 | Das Package java.util.regex | 563 |
| C.6 | Das Package java.io | 564 |
| C.7 | Das Package java.net | 567 |
| D | Spickzettel | 569 |
| D.1 | Listen | 569 |
| D.2 | Closures | 570 |
| D.3 | Reguläre Ausdrücke | 570 |
| D.4 | Unit Testing | 572 |
| D.5 | Mocks und Stubs | 572 |
| D.6 | XML-GPath-Notation | 573 |
| | Register | 575 |