

Eberhard Wolff

3., überarbeitete und erweiterte Auflage

dpunkt.verlag

1	Einleitung	1
1.1	Warum Spring? . . . . .	1
1.2	Was ist Spring? . . . . .	1
1.3	Spring und Java EE . . . . .	3
1.4	Woher kommt Spring? . . . . .	5
1.5	Warum dieses Buch? . . . . .	6
1.6	Patterns und dieses Buch. . . . .	6
1.7	Wie man das Buch lesen sollte. . . . .	7
1.8	Mit Spring entwickeln. . . . .	9
1.9	Danksagung . . . . .	9
2	Objekte finden zueinander: Dependency Injection	11
2.1	Übersicht . . . . .	11
2.2	Die Beispielanwendung . . . . .	12
2.2.1	Das fachliche Modell. . . . .	12
2.2.2	Geschäftsprozesse in der Beispielanwendung . . . . .	14
2.2.3	Benutzeroberfläche. . . . .	14
2.3	Objektnetze in OO-Systemen . . . . .	14
2.3.1	Netze weben. . . . .	15
2.4	Der neue Ansatz: Dependency Injection. . . . .	20
2.5	Dependency Injection mit Spring . . . . .	21
2.5.1	Die Konfigurationsdatei. . . . .	21
2.5.2	Die BeanFactory. . . . .	26
2.5.3	Constructor Dependency Injection . . . . .	27
2.5.4	Erzeugung mit Factories. . . . .	28

2.6	Vorteile von DI	31
2.6.1	Wir rufen Sie an	31
2.6.2	Flexibilität	32
2.6.3	Eingebaute Konfigurierbarkeit	32
2.6.4	Das Singleton-Pattern	33
2.6.5	Das Factory-Pattern	35
2.7	Die Grenzen von Dependency Injection	36
2.8	Wir kennen uns: Autowiring	39
2.9	Die bessere BeanFactory: Der ApplicationContext	42
2.9.1	Zugriff auf Ressourcen	43
2.9.2	Events	45
2.9.3	Internationalisierung	47
2.9.4	Objekte nachbearbeiten	48
2.10	Fortgeschrittene Techniken	51
2.10.1	Komplexe Datentypen als Werte setzen	51
2.10.2	Wie Properties genau konfiguriert werden	54
2.10.3	Properties mit ConversionService konvertieren	58
2.10.4	Lebenszyklus	59
2.10.5	Fachliche Klassen mit @Configurable konfigurieren	62
2.10.6	Konfigurationen handhaben	65
2.10.7	Spring Expression Language	72
2.10.8	Dynamische Sprachen mit Spring	73
2.11	Dependency Injection mit Annotationen	74
2.12	Definition von Spring-Beans mit Annotationen	78
2.13	JSR 330 Dependency-Injection-Annotationen	81
2.14	Spring Konfiguration mit Java-Klassen	83
2.15	Tests mit Spring	85
2.15.1	Spring und Unit-Tests	86
2.15.2	Integrationstests	90
2.15.3	Funktionale Tests	93
2.15.4	Systemtests	95
2.16	Annotationen oder XML?	95
2.17	Fazit	96

3	Aspektororientierte Programmierung mit Spring	99
3.1	Übersicht	99
3.2	Was sind Aspekte?	100
3.3	Aspekte mit Spring: Grundlagen	101
3.4	Spring AOP eigene Syntax	108
3.4.1	Pointcuts in Spring	108
3.4.2	Aspekte-Werkzeugkasten	111
3.5	AspectJ-Syntax mit Spring AOP	113
3.5.1	Verwendung klassischer Spring-Aspekte mit AspectJ-Pointcuts	113
3.5.2	Aspekte mit Annotationen markieren	115
3.5.3	Mehr über AspectJ Pointcuts	121
3.5.4	Interfaces nachträglich implementieren mit AspectJ-Syntax	123
3.6	Fortgeschrittene Techniken mit Spring-Aspekten	126
3.6.1	Reihenfolge der Advices	126
3.6.2	Das Advised-Interface und Zugriff auf den Proxy	126
3.6.3	Zielobjekte austauschen	127
3.7	Aspekte in der Beispielanwendung	132
3.8	AspectJ	137
3.8.1	Migration von Spring AOP nach AspectJ	138
3.8.2	AspectJ Load Time Weaving mit Spring	139
3.8.3	AspectJ-Aspekte mit Spring konfigurieren	140
3.9	Fazit	141
4	Transaktionen	143
4.1	Übersicht	143
4.2	Der Transaktionsmanager	144
4.2.1	Was ist eine Transaktion?	144
4.2.2	Die Pattern-Form	147
4.2.3	Das Exception-Übersetzer-Pattern	148
4.2.4	Die TransactionException-Hierarchie	149
4.3	Transaktionen selbst managen	150
4.4	Transaktionen mit Templates verwalten	153
4.4.1	Das Template-Pattern	153
4.4.2	Das Transaction-Template	154
4.5	Deklaratives Transaktionsmanagement	156

## Inhaltsverzeichnis

4.6	Transaktionen mit Spring in der Praxis . . . . .	160
4.7	Fazit . . . . .	162
5	Persistenz	163
5.1	Übersicht . . . . .	163
5.2	Das DAO-Pattern . . . . .	164
5.3	JDBC-Helferklassen . . . . .	167
5.3.1	Die JdbcTemplate-Klasse . . . . .	167
5.3.2	NamedParameterJdbcTemplate . . . . .	173
5.3.3	SimpleJdbcTemplate . . . . .	174
5.3.4	Anfragen als Objekte . . . . .	175
5.4	Datenbank-Exceptions . . . . .	179
5.5	iBatis-Unterstützung . . . . .	180
5.6	Hibernate-Integration . . . . .	184
5.7	JPA-Integration . . . . .	189
5.8	JDO . . . . .	195
5.9	Fazit . . . . .	196
6	Spring ins Netz	197
6.1	Übersicht . . . . .	197
6.2	RMI . . . . .	198
6.2.1	Das Exporter-Pattern . . . . .	199
6.2.2	Der RmiServiceExporter . . . . .	200
6.2.3	RMI-Objekte verwenden . . . . .	201
6.2.4	Das Proxy-Pattern . . . . .	201
6.2.5	Die RmiFactoryBean . . . . .	202
6.3	HTTP-basierte Protokolle . . . . .	203
6.4	SOAP . . . . .	207
6.4.1	SOAP mit Spring Web Services . . . . .	207
6.4.2	Andere SOAP-Ansätze . . . . .	214
6.5	EJB . . . . .	215
6.5.1	Was ist EJB? . . . . .	215
6.5.2	EJB 2.1 Beans mit Spring . . . . .	216
6.5.3	EJB 3 . . . . .	223
6.6	<i>f</i> Infrastrukturen: EJB, Spring . . . . .	225
6.7	Fazit . . . . .	228

7	Spring-Webtechnologien	231
7.1	Übersicht	231
7.2	Was ist MVC 2?.	232
7.3	Controller	233
7.3.1	Der AbstractController.	234
7.3.2	MultiActionController: Mehrere Controller in einem.	235
7.3.3	SimpleFormController für Formulare.	238
7.3.4	ExceptionHandler.	240
7.3.5	Controller mit Annotationen.	241
7.3.6	Fazit: Welcher Controller soll's sein?.	247
7.4	Validierung	248
7.5	Validierung mit JSR 303.	250
7.6	JSPs mit der Spring Tag Library.	253
7.7	Alternative View-Technologien.	260
7.7.1	XSLT-Views mit Spring	260
7.7.2	Excel-Views.	264
7.7.3	PDF-Views.	265
7.7.4	Weitere View-Technologien.	267
7.8	Welche View wird genutzt?.	268
7.8.1	InternalResourceViewResolver.	269
7.8.2	XmlViewResolver.	269
7.8.3	Mehrere ViewResolver nutzen.	270
7.8.4	Optimierungen	271
7.9	Handler Mapping	271
7.10	Der Weg eines Requests durch Spring MVC.	274
7.11	REST mit Spring	276
7.12	Spring und andere Webtechnologien.	282
7.12.1	JavaServer Faces (JSF).	282
7.12.2	Andere Technologien.	285
7.13	Scopes für Webanwendungen.	287
7.14	Spring Web Flow.	288
7.15	Spring JavaScript	296
7.16	Spring Faces.	297
7.17	Fazit	297

8	Weitere Enterprise-Features	299
8.1	Übersicht	299
8.2	Mit Sicherheit: Spring Security	300
8.3	JMS-Integration	311
8.4	Java Connector Architecture (JCA)	320
8.5	E-Mail-Unterstützung	322
8.6	Zeitgesteuerte Aufgaben	324
8.6.1	JDK-Timer	324
8.6.2	Quartz	327
8.7	Spring Dynamic Modules for OSGi Service Platforms	329
8.8	SpringSource dm Server	336
8.9	Management: JMX	346
8.10	Management: Spring Enterprise	349
8.11	Performance	351
8.12	Fazit	355
9	Andere Spring-Projekte	357
9.1	Übersicht	357
9.2	Entwicklungswerkzeuge	357
9.2.1	Spring-IDE	358
9.2.2	SpringSource Tool Suite	360
9.2.3	Dokumentieren mit BeanDoc	362
9.3	Spring Modules	363
9.4	Fazit	366
10	Fazit: Was bringt's?	367
11	Bibliografie	371
	Index	385