



Inhalt

Vorwort und Einleitung	5
Teil 1: Java-Grundlagen	15
1 Generics	17
1.1 Einleitung und Motivation	17
1.2 Erste Beispiele	19
1.3 Mehrere Typparameter	22
1.4 Einschränkung parametrisierter Typen	22
1.5 Umsetzung des Generics-Konzepts	23
1.6 Typenkompatibilität und Wildcards	25
1.6.1 Rohe Typen	27
1.6.2 Wildcards	27
1.7 Vererbung	30
1.8 Generische Methoden	32
1.9 Überladen und Überschreiben	35
1.10 Fazit	38
2 Reflection	39
2.1 Grundlagen von Reflection	40
2.1.1 Die Klasse Class	40
2.1.2 Die Klasse Field	42
2.1.3 Die Klasse Method	43
2.1.4 Die Klasse Constructor	44
2.1.5 Beispiel	45
2.1.6 Anwendungen	46
2.2 Reflection mit Generics	48
2.2.1 Reflection-Typsystem	49
2.2.2 Zusätzliche Methoden in Reflection-Klassen	51
2.2.3 Beispiel	51

3	Annotationen	57
3.1	Deklaration und Nutzung von Annotationen	57
3.2	Meta-Annotationen	60
3.3	Regeln für Annotationsdeklarationen	62
3.4	Reflection für Annotationen	63
3.5	Beispiel	67
3.6	Anwendung: Dependency Injection	69
4	Dynamische Proxies	73
4.1	Statische Proxies	74
4.2	Erzeugung dynamischer Proxies mit Reflection	79
4.3	Erzeugung dynamischer Proxies mit CGLIB	83
5	ClassLoader und Hot Deployment	88
5.1	Klassenladen in Java	88
5.2	ClassLoader	90
5.3	Beispiele	92
5.3.1	Hot Deployment	92
5.3.2	Gleichzeitige Nutzung mehrerer Versionen einer Klasse	94
Teil 2:	Java-Komponenten	99
6	Prototypische Implementierung eines Komponentensystems	101
6.1	Beispielkomponenten	102
6.1.1	Komponente Nr. 1	102
6.1.2	Komponente Nr. 2	105
6.1.3	Komponente Nr. 3	107
6.1.4	Rückblick	110
6.1.5	Variation der Komponentenbeispiele	111
6.2	Framework	113
6.2.1	Struktur des Komponenten-Frameworks	113
6.2.2	Die Klasse ComponentManager	115
6.2.3	Die Klasse DeploymentDirectoryListener	120
6.2.4	Restliche Klassen	121
7	Komponenten und Komponentensysteme	123
7.1	Modularität als grundlegendes Prinzip von Komponentensystemen	123
7.2	Definitionen für Software-Komponenten	125
7.3	Eigenschaften von Java-Komponenten	127
7.4	Beispiele und Gegenbeispiele für Komponentensysteme	130
7.4.1	Beispiele aus dem Java-Umfeld	130
7.4.2	Gegenbeispiele aus dem Java-Umfeld	131
7.4.3	Beispiele aus dem Nicht-Java-Umfeld	131

Teil 3: Beispiele für Java-Komponentensysteme	133
8 Java Beans	135
8.1 Komponentenmodell	135
8.2 Gebundene Eigenschaften und Eigenschaften mit Vetomöglichkeit	137
8.3 BeanInfo	140
8.4 Software-Werkzeuge	142
8.5 Bewertung	143
9 OSGi	145
9.1 Komponentenmodell	145
9.2 Erstes Beispiel-Bundle	147
9.3 Zweites Beispiel-Bundle	151
9.4 Variationen der Beispiel-Bundles	154
9.5 Hot Deployment	156
9.6 Lebenszyklus von Komponenten	160
9.7 BundleContext und Bundle	161
9.8 Erweiterungen von OSGi	162
9.8.1 Declarative Services	162
9.8.2 Zusätzliche Erweiterungen	166
9.9 Versionen von Komponenten	167
9.10 Bewertung	168
10 Eclipse	170
10.1 Architektur von Eclipse	171
10.1.1 Eclipse-Funktionsgruppen	171
10.1.2 Workspace und Workbench	172
10.1.3 Erweiterungspunkte (Extension Points)	173
10.2 Komponentenmodell von Eclipse	174
10.3 Erstes Eclipse-Plugin	175
10.4 Weitere Eclipse-Plugins	181
10.5 Erweiterung der Eclipse-Plugins	182
10.6 Klassenladen bei Bedarf	183
10.7 Bewertung	185
11 Applets	186
11.1 Erstes Beispiel	186
11.2 Zweites Beispiel	189
11.3 Bewertung	192

12	Servlets	193
12.1	Verzeichnisstruktur eines Web-Servers	193
12.2	Komponentenmodell	194
12.2.1	Verzeichnisstruktur einer Komponente	194
12.2.2	Die Konfigurationsdatei web.xml	196
12.2.3	Java-Code einer Web-Komponente	197
12.3	Erste Beispielkomponente	201
12.4	Zweite Beispielkomponente	205
12.5	Bewertung	210
13	Enterprise Java Beans (EJB)	212
13.1	Mehrschichtige Architekturen	212
13.2	Interaktion mit EJB-Komponenten	214
13.3	Klassenarten einer EJB-Komponente	216
13.4	Session Beans	216
13.4.1	Stateful Session Beans	216
13.4.2	Stateless Session Beans	217
13.4.3	Singleton Session Beans	218
13.5	Komponentenmodell	219
13.6	Erste EJB-Beispielkomponente	220
13.6.1	Server-Seite	220
13.6.2	Client-Seite	222
13.7	Zweite EJB-Beispielkomponente	225
13.8	Dritte EJB-Beispielkomponente (Call-By-Value)	228
13.9	Vierte EJB-Beispielkomponente (Call-By-Reference)	230
13.9.1	Getrennte EJB-Jar-Dateien	230
13.9.2	Gemeinsame Ear-Datei mit Dependency Injection	233
13.9.3	Lokale Schnittstellen	234
13.10	Entities und Transaktionssteuerung	235
13.11	Funktionen eines EJB-Containers	238
13.12	Bewertung	239
14	Spring	240
14.1	Komponentenmodell	240
14.2	Erste Spring-Anwendung: Singletons und Prototypes	242
14.3	Zweite Spring-Anwendung: Dependency Injection	244
14.4	Factory-Methoden und Factory-Beans	246
14.5	Autowiring	246
14.6	Dritte Spring-Anwendung: Konfiguration durch Annotationen	247
14.7	Vierte Spring-Anwendung: BeanPostProcessor	248
14.8	Aspektororientierte Programmierung (AOP)	251

14.9	Fünfte Spring-Anwendung: AOP	252
14.10	Bewertung	255
15	Ereignisbusse	257
15.1	Grundkonzepte von Ereignisbussen	257
15.2	Komponentenmodell von RRIbbit	259
15.3	Erste RRIbbit-Anwendung	261
15.4	Zweite RRIbbit-Anwendung	264
15.5	Bewertung	267
16	Android	269
16.1	Software-Architektur von Android	270
16.2	Prinzipien der Ausführung von Apps	271
16.3	Komponentenmodell	275
16.4	Anwendung mit einer Activity	277
16.4.1	Activity mit programmierter Oberfläche	277
16.4.2	Activity mit XML-definierter Oberfläche	279
16.5	Anwendung mit mehreren Activities	282
16.5.1	Start einer Activity mit explizitem Intent	282
16.5.2	Start einer Activity mit implizitem Intent	284
16.5.3	Activity mit Resultat	285
16.5.4	Variationen	287
16.6	Lebenszyklus von Activities	288
16.7	Service und Activity im Vergleich	290
16.8	Anwendung mit einem ungebundenen Service	291
16.9	Anwendung mit einem gebundenen Service	294
16.9.1	AIDL-Schnittstelle	294
16.9.2	Implementierung einer AIDL-Schnittstelle	294
16.9.3	Realisierung eines gebundenen Service	295
16.9.4	Nutzung eines gebundenen Service	296
16.9.5	Parameterübergabe durch Call-By-Value-Result	298
16.9.6	Parameterübergabe durch Call-By-Reference	300
16.10	Bewertung	303
	Literatur	305
	Index	307