

# Inhaltsverzeichnis

<b>Vorwort</b>	<b>XI</b>
<b>1 Java 6 Basics.....</b>	<b>1</b>
1.1 Enumerationen.....	2
1.2 Import statischer Member.....	4
1.2.1 Das Ende eines Anti-Patterns.....	4
1.3 Interfaces vs. Klassen.....	5
1.3.1 Typ vs. Implementierung.....	7
1.4 for-each Schleife.....	8
1.5 Variable Anzahl von Argumenten.....	10
1.6 Autoboxing .....	12
1.7 Überblick über Mustang-Features.....	13
1.8 Neue I/O-Methoden.....	14
1.8.1 File.....	15
1.8.2 Console-I/O.....	17
1.9 Floating-Point Essentials.....	19
1.9.1 Floating-Points vs. Reelle Zahlen.....	19
1.9.2 Neue Konstante und Methoden in Java 6.....	24
1.9.3 Floating-Point Constraints .....	27
1.10 Typsichere Array-Kopien.....	29
1.11 Erweiterung des Collection-APIs.....	32
1.11.1 API-Überblick.....	32
1.11.2 Utility-Klasse Collections.....	36
1.11.3 Queue, Deque und ConcurrentStack.....	38
1.11.4 Konsistenz von hashCode(), equals() und compare().....	43
1.11.5 Navigierbare Container.....	49
1.11.6 ConcurrentSkipListSet.....	53
1.11.7 ConcurrentMap.....	55
1.11.8 CopyOnWrite.....	56

1.12	<b>Fazit.....</b>	58
1.13	<b>Referenzen.....</b>	58
<b>2</b>	<b>Generische Techniken.....</b>	<b>59</b>
2.1	<b>Generics Grundlagen.....</b>	61
2.1.1	Generisches Anti-Pattern.....	61
2.1.2	Klassische Design-Prinzipien .....	63
2.1.3	Generics vs. Object.....	65
2.1.4	Eine einfacher Typ-Variable <T>.....	65
2.1.5	Einschränkungen für <T>.....	68
2.1.6	Generische Methoden.....	71
2.1.7	Typ-Inferenz .....	73
2.2	<b>Arrays und Typ-Hierarchien mit Generics.....</b>	<b>75</b>
2.2.1	Arrays vs. generischen Listen.....	75
2.2.2	Invarianz.....	77
2.3	<b>Wichtige generische Begriffe.....</b>	<b>80</b>
2.4	<b>Restriktionen zu Generics.....</b>	<b>84</b>
2.4.1	Unerlaubte generische Ausdrücke .....	84
2.5	<b>Finden von Typ-Einschränkungen.....</b>	<b>88</b>
2.5.1	Generics, ein statisches Konzept.....	92
2.6	<b>Wildcard-Ausdrücke.....</b>	<b>94</b>
2.6.1	Syntax und Eigenschaften von Wildcards.....	94
2.6.2	Ungebundene Wildcards.....	96
2.6.3	Wildcards mit extends.....	100
2.6.4	Wildcards mit super.....	102
2.6.5	Finden aktueller Typen mit Wildcards.....	104
2.6.6	Wildcard Capture.....	106
2.7	<b>Covariant Return beim Overriding.....</b>	<b>107</b>
2.8	<b>Overriding mit covarianten Parameter-Typen.....</b>	<b>109</b>
2.9	<b>Overloading bei gleicher Signatur.....</b>	<b>113</b>
2.10	<b>Type-Token Varianten.....</b>	<b>114</b>
2.10.1	Class-Token Idiom.....	115
2.10.2	Anlage eines Arrays zum aktuellen Typ.....	118
2.11	<b>Das Self-Type Idiom.....</b>	<b>120</b>
2.11.1	Self-Types im Einsatz.....	120
2.12	<b>Enumerationen.....</b>	<b>126</b>
2.12.1	EnumSet.....	129
2.12.2	EnumMap.....	130

2.12.3 Anonyme Klassen in enum.....	134
<b>2.13 FAQ.....</b>	<b>137</b>
<b>2.14 Fazit.....</b>	<b>147</b>
<b>2.15 Referenzen.....</b>	<b>148</b>
<b>3 Reflektive Techniken.....</b>	<b>149</b>
<b>3.1 Überblick.....</b>	<b>149</b>
3.1.1 Compilieren & Linken: Java vs. C/C++.....	150
3.1.2 Laden via ClassLoader.....	151
3.1.3 Sicherheit und Performanz.....	153
3.1.4 Zentrale Aufgaben von Reflexion.....	153
3.1.5 Reflexion-API ist read-only.....	154
<b>3.2 Klassen-basiertes Reflection-API.....</b>	<b>155</b>
3.2.1 Klassen-Modell.....	155
3.2.2 Ermitteln einer Class<T>-Instanz.....	156
<b>3.3 Class &amp; Member.....</b>	<b>158</b>
3.3.1 Warnungen unterdrücken: asSubClass() und cast().....	159
3.3.2 Neue Class-Methoden isX().....	162
3.3.3 Namenskonventionen zu Class-Methoden getX().....	165
3.3.4 Getter für Klassen/Interface-Namen .....	166
3.3.5 Getter für Klassen/Package-Beziehungen.....	167
3.3.6 Getter für Felder, enum-Konstante.....	170
3.3.7 Getter für Methoden.....	172
3.3.8 Getter für Konstruktoren & statische Methoden.....	174
3.3.9 Modifier, Member, AccessibleObject.....	176
3.3.10 Klasse Field.....	178
3.3.11 Klasse Constructor, Initialisierung.....	181
3.3.12 Klasse Method.....	184
3.3.13 Klasse Array.....	188
3.3.14 Problemfall primitiven Typen.....	192
<b>3.4 Interface-basiertes Reflexion-API.....</b>	<b>194</b>
3.4.1 Type .....	196
3.4.2 GenericDeclaration, TypeVariable, Field.....	196
3.4.3 Generische Methoden in Class, Method & Constructor.....	199
3.4.4 ParameterizedType, WildcardType, GenericArrayType.....	202
3.4.5 Super Type Token.....	206
<b>3.5 Fazit.....</b>	<b>211</b>
<b>3.6 Referenzen.....</b>	<b>211</b>

<b>4 Annotations-Techniken.....</b>	<b>213</b>
<b>4.1 Überblick.....</b>	<b>214</b>
4.1.1 Standard-Annotationen in Package <code>java.lang</code> .....	216
<b>4.2 Deklaration und Einsatz.....</b>	<b>219</b>
4.2.1 Restriktionen & Beispiele.....	221
<b>4.3 Meta-Annotationen.....</b>	<b>227</b>
<b>4.4 Allgemeine Standards.....</b>	<b>233</b>
4.4.1 Richtlinien für Annotations-Gültigkeit.....	234
<b>4.5 Core-Annotationen.....</b>	<b>235</b>
4.5.1 Defekt-sensitive Annotationen.....	235
4.5.2 Standard-Annotationen in Java 6.....	238
<b>4.6 Modifizieren bestehender Annotationen .....</b>	<b>240</b>
<b>4.7 Laufzeit-Zugriff auf Annotationen.....</b>	<b>241</b>
<b>4.8 Runtime-Auswertung von Annotationen.....</b>	<b>243</b>
<b>4.9 Validieren mit Hilfe von Annotationen.....</b>	<b>246</b>
4.9.1 Validierung, Test-Frameworks & AOP.....	246
4.9.2 Ein minimales Validierungs-Framework.....	247
4.9.3 Test und Analyse der Validierung.....	252
<b>4.10 Implementierung von Annotationen.....</b>	<b>254</b>
<b>4.11 Annotations-Prozessoren.....</b>	<b>259</b>
4.11.1 Annotation-Prozessor-API.....	260
4.11.2 Zugriff auf Programm-Elemente.....	265
4.11.3 Restriktionen für Prozessoren.....	267
4.11.4 Decorator-Style.....	268
4.11.5 Factory-Einsatz.....	269
4.11.6 Prozessor-generierte Validierung.....	270
<b>4.12 Annotation vs. Konfiguration.....</b>	<b>276</b>
<b>4.13 Fazit.....</b>	<b>276</b>
<b>4.14 Referenzen.....</b>	<b>277</b>
<b>5 Service &amp; Component-Techniken.....</b>	<b>279</b>
<b>5.1 ClassLoader-Techniken.....</b>	<b>280</b>
5.1.1 Namensräume.....	280
5.1.2 Struktur der Klasse ClassLoader.....	282
5.1.3 ClassLoader Delegation & Hierarchie.....	283
5.1.4 SecureClassLoader & Berechtigungen.....	286
5.1.5 URLClassLoader.....	289

---

5.1.6 Implizites/explizites Laden & Versions-Management.....	293
5.1.7 Thread Context-ClassLoaders.....	297
5.1.8 Child-First ClassLoader.....	302
5.1.9 Spezielle ClassLoader.....	307
5.1.10 Super-Packages und Module (Java 7).....	310
<b>5.2 Services – ein erster Überblick.....</b>	<b>313</b>
<b>5.3 Persistenz &amp; JDBC 4.....</b>	<b>314</b>
5.3.1 Persistenz-Alternativen.....	314
5.3.2 Persistenz mittels einer RDBMS.....	315
5.3.3 JDBC 4.....	317
5.3.4 Derby.....	318
5.3.5 Neue JDBC 4 Features.....	320
5.3.6 Query, DataSet<T>, @Select und @Update.....	321
<b>5.4 Das DAO-Pattern am Beispiel.....</b>	<b>327</b>
5.4.1 Beispiel Student-Kurs-Beziehung.....	327
5.4.2 DAO Design-Pattern.....	330
5.4.3 Generisches DAO-Interface.....	331
5.4.4 EoD mit generischen DAO.....	333
<b>5.5 Automatisches Laden von Services.....</b>	<b>344</b>
5.5.1 Klasse ServiceLoader.....	345
<b>5.6 Dynamisches Proxy.....</b>	<b>350</b>
5.6.1 Proxy API.....	350
5.6.2 Proxies als Adapter.....	356
5.6.3 Das Decorator-Pattern .....	363
5.6.4 Proxy als Decorator.....	365
5.6.5 Duck Typing – in Java.....	371
<b>5.7 JMX – Java Management Extensions.....</b>	<b>376</b>
5.7.1 JMX-Überblick.....	376
5.7.2 Arten von MBeans.....	377
5.7.3 Standard MBean.....	378
5.7.4 MXBeans.....	385
<b>5.8 Fazit.....</b>	<b>392</b>
<b>5.9 Referenzen.....</b>	<b>393</b>
<b>6 Compiler &amp; Scripting Techniken.....</b>	<b>395</b>
<b>6.1 Das Compiler-API.....</b>	<b>396</b>
6.1.1 Javac Mimikry.....	396
6.1.2 Überblick über das Package javax.tools.....	398
6.1.3 Compilierung mit Fehlern und Diagnose.....	401
6.1.4 Setzen der Pfade mit StandardLocation.....	403

6.1.5 DiagnosticCollector und String-Source.....	404
6.1.6 Der mühsame Weg zur In-Memory Class.....	407
<b>6.2 Das Scripting API.....</b>	<b>415</b>
6.2.1 Dynamisch vs. Statisch vs. Funktional.....	415
6.2.2 API Überblick.....	416
6.2.3 Eingebundene Skript-Sprachen.....	418
6.2.4 Einbinden von Skript-Sprachen.....	420
6.2.5 Ausführung von Skripts aus Java.....	421
6.2.6 Aufruf von Funktionen aus Java.....	425
6.2.7 Das Invocable Interface.....	427
6.2.8 Invocable an Beispielen.....	428
6.2.9 Das Compilable Interface.....	430
6.2.10 Java aus Skript-Sicht.....	432
6.2.11 JavaScript Anbindung an Java.....	432
6.2.12 JRuby Anbindung an Java.....	437
6.2.13 JavaFX Anbindung an Java.....	442
<b>6.3 Fazit.....</b>	<b>444</b>
<b>6.4 Referenzen.....</b>	<b>445</b>
<b>Index.....</b>	<b>447</b>