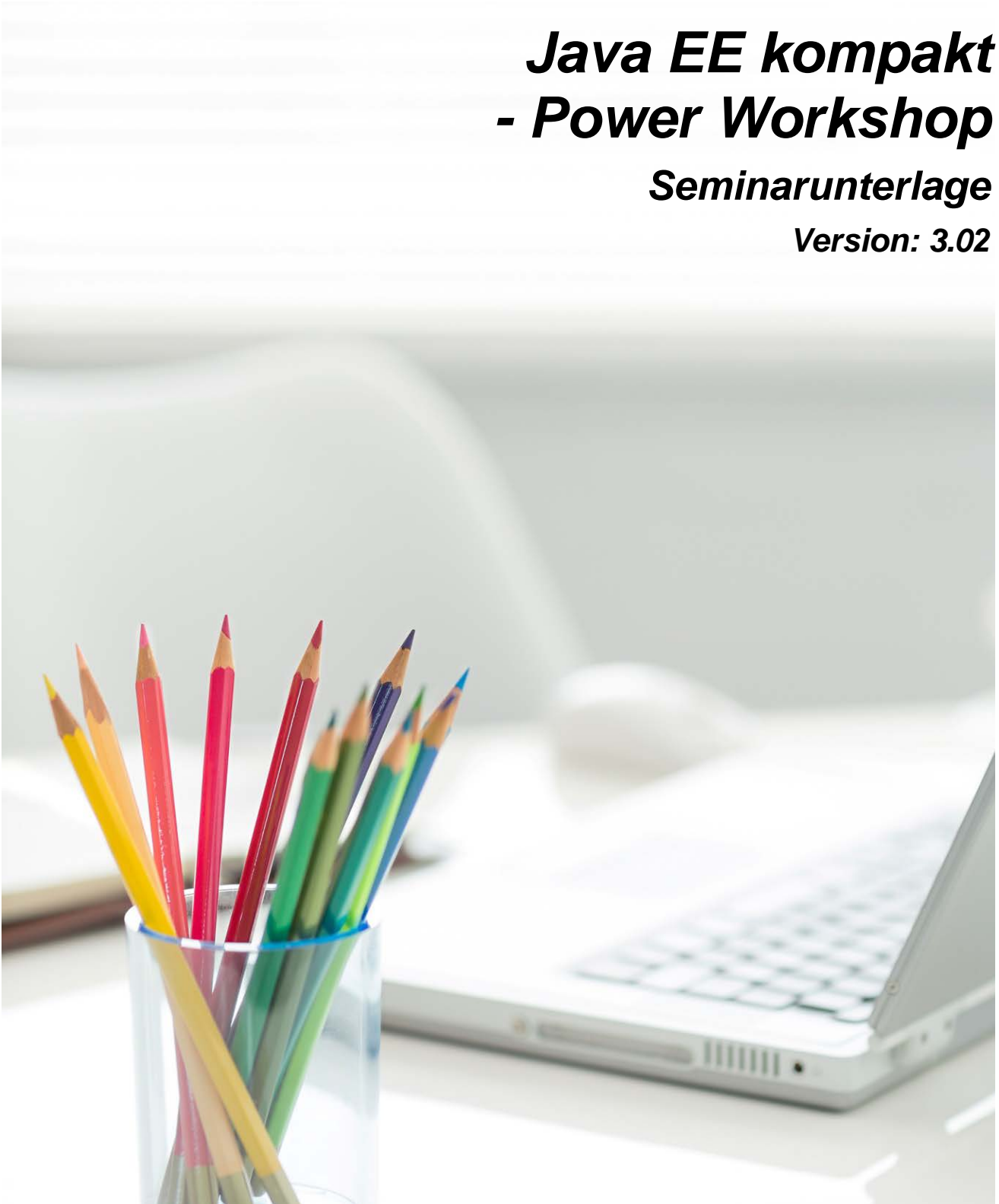


***Java EE kompakt
- Power Workshop***
Seminarunterlage
Version: 3.02



Dieses Dokument wird durch die ORDIX AG veröffentlicht.

Copyright ORDIX AG. Alle Rechte vorbehalten.

Alle Produkt- und Dienstleistungs-Bezeichnungen sind Warenzeichen oder eingetragene Warenzeichen der jeweiligen Firmen und beziehen sich auf Eintragungen in den USA oder USA-Warenzeichen.

Weitere Logos und Produkt- oder Handelsnamen sind eingetragene Warenzeichen oder Warenzeichen der jeweiligen Unternehmen.

Kein Teil dieser Dokumentation darf ohne vorherige schriftliche Genehmigung der ORDIX AG weitergegeben oder benutzt werden.

Adressen der ORDIX AG

Die ORDIX AG besitzt folgende Geschäftsstellen

ORDIX AG
Karl-Schurz-Straße 19a
D-33100 Paderborn
Tel.: (+49) 0 52 51 / 10 63 - 0
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG
An der alten Ziegelei 5
D-48157 Münster
Tel.: (+49) 02 51 / 9 24 35 – 00
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG
Welser Straße 9
D-86368 Gersthofen
Tel.: (+49) 08 21 / 507 492 – 0
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG
Kreuzberger Ring 13
D-65205 Wiesbaden
Tel.: (+49) 06 11 / 7 78 40 – 00
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG
Wikingerstraße 18-20
D-51107 Köln
Tel.: (+49) 02 21 / 8 70 61 – 0
Fax.: (+49) 01 80 / 1 67 34 90

Internet: <http://www.ordix.de>

Email: seminare@ordix.de

Inhaltsverzeichnis

| | | |
|----------|--|-----------|
| 1 | Java EE Plattform..... | 13 |
| 1.1 | Big Picture | 14 |
| 1.2 | „Typische“ Java EE Architektur..... | 15 |
| 1.3 | Java-EE-Architektur (4-Schichten-Architektur)..... | 16 |
| 1.4 | 4-Schichten-Architektur | 17 |
| 1.5 | Java EE – Plattform..... | 18 |
| 1.6 | Java-basierte Cross(Multi)-Plattform..... | 19 |
| 1.7 | Java-EE – Spezifikationen (JSR 342)..... | 20 |
| 1.8 | Java-EE-Applikationsserver..... | 21 |
| 1.9 | EJB-Container..... | 22 |
| 1.10 | Neuigkeiten..... | 23 |
| 1.10.1 | Java EE 8 Plattform | 23 |
| 1.10.2 | Java EE 8 – EE4J..... | 24 |
| 1.10.3 | JAKARTA EE..... | 25 |
| 1.11 | Easy Packaging | 26 |
| 1.11.1 | Artefakte (Auslieferung) | 26 |
| 1.12 | Easy packaging..... | 27 |
| 1.12.1 | Beispiel: Easy (War) packaging..... | 28 |
| 2 | Enterprise Java Beans..... | 29 |
| 2.1 | Enterprise Java Beans | 30 |
| 2.2 | Aufrufmodelle von Clients..... | 31 |
| 2.3 | Sessions Beans (SLSB, SFSB, SSB) | 32 |
| 2.3.1 | Mikroarchitektur einer Session-Bean | 32 |
| 2.3.2 | Bean-Klasse | 33 |
| 2.3.3 | Business-Interface | 34 |
| 2.3.4 | <i>No-Interface (Client) View</i> | 35 |
| 2.3.5 | Beispiel: Lokaler Zugriff - @EJB..... | 36 |
| 2.3.6 | Annotation @LocalBean | 37 |
| 2.3.7 | Bean-Instanz | 38 |
| 2.3.8 | Transaktionskontext..... | 39 |
| 2.3.9 | Beispiel: Deklarative Transaktionalität..... | 40 |
| 2.4 | Asynchrone Session Beans..... | 41 |
| 2.4.1 | Motivation..... | 41 |
| 2.4.2 | Asynchrone Methodenaufrufe | 42 |
| 2.4.3 | Asynchrone Session Bean | 43 |
| 2.4.4 | Clients (asynchrone) Kommunikation | 44 |
| 2.4.5 | AsyncResult - Nebenläufiger Task | 45 |
| 2.4.6 | Asynchrone (Stateful) Session Beans | 46 |
| 2.4.7 | Annotation @AccessTimeout..... | 47 |
| 2.5 | Stateless Session Beans (kein Client-Bezug) | 48 |
| 2.5.1 | Lebenszyklus..... | 48 |
| 2.5.2 | Instanz-Pooling..... | 49 |
| 2.5.3 | Verhalten bei Exceptions | 50 |
| 2.5.4 | Application Exceptions..... | 51 |
| 2.5.5 | SystemException und Bean LifeCycle..... | 52 |
| 2.6 | Stateful Session Beans (Client-Bezug) | 53 |
| 2.6.1 | Eigenschaften..... | 53 |
| 2.6.2 | Lebenszyklus..... | 54 |
| 2.6.3 | Verhalten bei Exception | 55 |
| 2.6.4 | Beenden der Konversation..... | 56 |
| 2.6.5 | Passivierung..... | 57 |
| 2.6.6 | Passivierungskandidaten | 58 |
| 2.6.7 | Automatische Freigabe – Timeout..... | 59 |
| 2.6.8 | Interface SessionSynchronization | 60 |
| 2.7 | Singleton..... | 61 |
| 2.7.1 | Singleton-Pattern..... | 61 |
| 2.7.2 | Singleton Session Bean..... | 62 |

| | | |
|----------|--|-----------|
| 2.7.3 | Lebenszyklus..... | 63 |
| 2.7.4 | Initialisierung..... | 64 |
| 2.7.5 | Initialisierung – Abhängigkeiten..... | 65 |
| 2.7.6 | Nebenläufigkeit..... | 66 |
| 2.7.7 | <i>Container-Managed</i> Nebenläufigkeit..... | 67 |
| 2.7.8 | Singleton Session Bean - Loopback-Calls..... | 68 |
| 2.7.9 | Session-Bean – Fazit..... | 69 |
| 3 | EJB-Container Dienste | 70 |
| 3.1 | Zugriff auf EJBs über JNDI..... | 71 |
| 3.1.1 | Namen und Verzeichnisdienst..... | 71 |
| 3.1.2 | Standardisierung - Portable JNDI-Namen..... | 72 |
| 3.1.3 | JNDI-Namen – Besonderheiten..... | 73 |
| 3.1.4 | Beispiel 1: JNDI Zugriff auf EJB – Single View..... | 74 |
| 3.1.5 | Beispiel 2: JNDI Zugriff auf EJB – Multiple Views..... | 75 |
| 3.1.6 | Beispiel 3: DI mit Portable JNDI-Namen..... | 76 |
| 3.1.7 | Zugriff auf JNDI – Clientseitig..... | 77 |
| 3.2 | Timer Service..... | 78 |
| 3.2.1 | Timer Service..... | 78 |
| 3.2.2 | Programmgesteuerte Timer..... | 79 |
| 3.2.3 | Timeout Methode..... | 80 |
| 3.2.4 | Persistente Timer..... | 81 |
| 3.2.5 | Automatische Timer..... | 82 |
| 3.2.6 | Zeitpläne..... | 83 |
| 3.2.7 | Beispiel: Zeitpläne..... | 84 |
| 3.2.8 | Beispiel: Programmgesteuerter Timer..... | 85 |
| 3.2.9 | Beispiel: Calendar Based Timer..... | 86 |
| 3.2.10 | Beispiel: Calendar Based Timer Service..... | 87 |
| 3.3 | Interzeptoren..... | 88 |
| 3.3.1 | Aspektororientierte Programmierung (AOP)..... | 88 |
| 3.3.2 | Interzeptoren in Java-EE..... | 89 |
| 3.3.3 | Interzeptor-Typen..... | 90 |
| 3.3.4 | Interzeptoren in der EJB-Spezifikation..... | 91 |
| 4 | Message-Driven Beans..... | 94 |
| 4.1 | Java EE Messaging..... | 95 |
| 4.1.1 | Motivation..... | 95 |
| 4.1.2 | Java Messaging Service (JMS)..... | 96 |
| 4.1.3 | Messaging-Infrastruktur..... | 97 |
| 4.1.4 | Message-Broker..... | 98 |
| 4.2 | Java Messaging Service – JMS 1.1 API..... | 99 |
| 4.2.1 | JMS 1.1 API Programmiermodell..... | 99 |
| 4.2.2 | Queue - <i>Point-to-Point(P2P)</i> Messaging..... | 100 |
| 4.2.3 | Queue-Konfiguration..... | 101 |
| 4.2.4 | Topic - <i>Publish/Subscribe</i> Messaging..... | 102 |
| 4.2.5 | Topic-Konfiguration..... | 103 |
| 4.2.6 | JMS Interfaces..... | 104 |
| 4.2.7 | JMS-Message..... | 105 |
| 4.2.8 | JMS-Message – <i>Headers</i> | 106 |
| 4.2.9 | JMS-Message – <i>Payload</i> | 107 |
| 4.2.10 | JMS-Message – <i>Properties</i> | 108 |
| 4.2.11 | JMS-Message – <i>Selectors</i> | 109 |
| 4.2.12 | JMS-Message – Persistente Nachrichten..... | 110 |
| 4.2.13 | JMS-Message – Priorität..... | 111 |
| 4.2.14 | JMS-Message – <i>ExpiredDate</i> | 112 |
| 4.2.15 | JMS-Message - Dauerhafte Nachrichten..... | 113 |
| 4.2.16 | Client-Zugriff, Details..... | 114 |
| 4.3 | Neuerungen in JMS 2.0..... | 118 |
| 4.3.1 | Neuerungen in JMS 2.0..... | 118 |
| 4.3.2 | JMS 2.0 Programmiermodell..... | 119 |

| | | |
|----------|--|------------|
| 4.3.3 | AutoCloseable | 120 |
| 4.3.4 | Neue JMS 2.0 API | 121 |
| 4.3.5 | JMS Message Versand mit neuer API | 122 |
| 4.3.6 | Injection des JMSContext..... | 123 |
| 4.3.7 | Versand mit injiziertem – JMSContext..... | 124 |
| 4.3.8 | Neue API Features | 125 |
| 4.3.9 | JMS Ressourcen in Java EE..... | 127 |
| 4.3.10 | Sonstige Neuerungen | 128 |
| 4.3.11 | Zusammenfassung JMS 2.0..... | 130 |
| 4.4 | Message Driven Beans | 131 |
| 4.4.1 | JMS und Java EE | 131 |
| 4.4.2 | Java EE Messaging | 132 |
| 4.4.3 | Message-Driven Beans..... | 133 |
| 4.4.4 | MDB (EJB 3.0)..... | 134 |
| 4.4.5 | MDB (EJB 3.2)..... | 135 |
| 4.4.6 | Beispiel: JMS-EJB-Client | 136 |
| 5 | Contexts and Dependency Injection (Java EE platform)..... | 137 |
| 5.1 | Motivation | 138 |
| 5.2 | Dependency Injection (DI)..... | 139 |
| 5.3 | Warum CDI in Java EE?..... | 140 |
| 5.4 | CDI in Java EE 6 (JSR-330)..... | 141 |
| 5.5 | CDI in Java EE (JSR-299, JSR-330) | 142 |
| 5.6 | Annotationen..... | 143 |
| 5.7 | Schichtenübergreifende Injektion..... | 144 |
| 5.8 | CDI Scope | 145 |
| 5.9 | Der Begriff „Bean“ in CDI..... | 146 |
| 5.10 | Beispiel: Managed Bean 1.0 - CDI Bean..... | 147 |
| 5.11 | Der Begriff „Bean“ in Java EE - EJB, REST, CDI..... | 148 |
| 5.12 | CDI - Architektur (JSR-299, JSR-330) | 149 |
| 5.13 | CDI Container - Spezifische Dienste..... | 150 |
| 5.14 | Beispiel: CDI-Bean..... | 151 |
| 5.15 | Managed Beans - Spezifische Annotationen..... | 152 |
| 5.16 | Classpath - beans.xml..... | 153 |
| 5.17 | Deklarationen in beans.xml | 154 |
| 5.18 | Basis-Injection in CDI | 155 |
| 5.19 | Beispiel 1: <i>Feld-Injektion</i> | 156 |
| 5.20 | Beispiel 2: <i>Konstruktor-Injektion</i> | 157 |
| 5.21 | Beispiel 3: <i>Setter-Injektion</i> | 158 |
| 5.22 | Qualifier | 159 |
| 5.23 | Definition Qualifier | 160 |
| 5.24 | Beispiel: Schritt #1 - #3..... | 162 |
| 5.25 | <i>Built-In Qualifiers</i> | 163 |
| 5.26 | @Named - Bean in EL | 164 |
| 5.27 | @Any..... | 165 |
| 5.28 | @New..... | 166 |
| 5.29 | Interface Instance<T> | 167 |
| 5.30 | Beispiel | 168 |
| 5.31 | Stateful-Objekte | 169 |
| 5.32 | Built-In Scope - Stateful-Objekte..... | 170 |
| 5.33 | Dependent Scope (Default)..... | 171 |
| 5.34 | Request Scope | 172 |
| 5.35 | Session Scope | 173 |
| 5.36 | @ConversationScoped | 174 |
| 5.37 | Conversation Scope..... | 175 |
| 5.38 | CDI in Java SE..... | 176 |
| 5.39 | Erzeugung von Bean Instanzen - @Produces | 177 |
| 5.40 | Beispiel: Consumer – Producer | 178 |
| 5.41 | Erzeugung anderer Instanzen - @Produces | 179 |
| 5.42 | Typsichere Java-EE-Ressourcen - Producer (I)..... | 180 |

| | | |
|----------|---|------------|
| 5.43 | Typsichere Java-EE-Ressourcen - Consumer (II)..... | 181 |
| 5.44 | @Disposes | 182 |
| 5.45 | CDI Events – Prinzip | 183 |
| 5.46 | CDI Events – POJOs..... | 184 |
| 5.47 | Beispiel 1 | 185 |
| 5.48 | Beispiel 2 | 186 |
| 5.49 | Alternative Bean..... | 187 |
| 5.50 | Beispiel | 188 |
| 5.51 | Interceptors..... | 189 |
| 5.52 | CDI Interzeptoren..... | 190 |
| 5.53 | Aktivierung der CDI Interzeptoren..... | 191 |
| 5.54 | Composite Interceptor Binding..... | 192 |
| 5.55 | CDI Decorator | 193 |
| 5.56 | Stereotypen | 194 |
| 5.57 | Beispiel: Ohne Stereotypes | 195 |
| 5.58 | Beispiel: Einsatz von Stereotypes..... | 196 |
| 5.59 | Built-In Stereotypes in JSF- @Model | 197 |
| 6 | Best Practices | 198 |
| 6.1 | Komponenten der EJB-Anwendung..... | 199 |
| 6.2 | Facade..... | 200 |
| 6.3 | Dependency Injection..... | 201 |
| 6.4 | Beispiel Fassade..... | 202 |
| 6.5 | Interzeptoren..... | 203 |
| 6.6 | Business Methoden Interzeptoren | 204 |
| 6.7 | Service Bean..... | 205 |
| 6.8 | Beispiel Service Bean | 206 |
| 6.9 | Data Access Layer | 207 |
| 7 | Java Persistence API | 208 |
| 7.1 | Motivation | 209 |
| 7.2 | Persistenz Framework in Java..... | 210 |
| 7.3 | JPA Timeline..... | 211 |
| 7.4 | JPA 1.0 | 212 |
| 7.5 | JPA 2.0..... | 213 |
| 7.6 | JPA 2.1..... | 214 |
| 7.7 | JPA 2.2..... | 215 |
| 7.8 | JPA - Grundlagen..... | 216 |
| 7.8.1 | JPA Provider (Anbieter) | 216 |
| 7.8.2 | Beispiel: JPA Provider..... | 217 |
| 7.8.3 | DB Konfiguration via XML – <i>persistence.xml</i> | 218 |
| 7.8.4 | Java Persistenz API..... | 219 |
| 7.8.5 | EntityManager | 220 |
| 7.8.6 | Persistence Context..... | 221 |
| 7.8.7 | Typen von EntityManager | 222 |
| 7.8.8 | Java SE EntityManager..... | 223 |
| 7.8.9 | Entity Manager – Transaktionalität | 224 |
| 7.8.10 | EntityTransaction | 225 |
| 7.8.11 | Java EE EntityManager..... | 226 |
| 7.8.12 | EntityManager Interface..... | 227 |
| 7.8.13 | Beispiel: Arbeitsweise des Entity Manager | 228 |
| 7.8.14 | Entity Lebenszyklus (Zustände) | 229 |
| 7.8.15 | Detachment | 231 |
| 7.8.16 | JPA Konfiguration (Persistence Unit)..... | 232 |
| 7.8.17 | Persistence Konfiguration | 233 |
| 7.8.18 | Beispiel 1: Derby..... | 235 |
| 7.8.19 | Beispiel 2: hsqldb..... | 236 |
| 7.8.20 | Beispiel 3: MySQL | 237 |
| 7.8.21 | Entity | 238 |
| 7.9 | ORM – simple Mapping..... | 240 |

| | | |
|--------|---|-----|
| 7.9.1 | ORM - JPA Annotationen..... | 240 |
| 7.9.2 | Default Mapping..... | 241 |
| 7.9.3 | Zu persistierende Felder | 242 |
| 7.9.4 | @Basic..... | 243 |
| 7.9.5 | Beispiel: @Basic..... | 245 |
| 7.9.6 | @Transient..... | 246 |
| 7.9.7 | Beispiel: @Transient..... | 247 |
| 7.9.8 | Entity Identität – Primärschlüssel..... | 248 |
| 7.9.9 | Einfacher Primärschlüssel @Id | 249 |
| 7.9.10 | Primärschlüssel - Generierte PK | 250 |
| 7.9.11 | Beispiel: GenerationType.AUTO | 251 |
| 7.9.12 | Beispiel: GenerationType.SEQUENCE | 252 |
| 7.9.13 | Komplexer Primärschlüssel..... | 253 |
| 7.9.14 | Primärschlüssel - @IdClass | 254 |
| 7.9.15 | Beispiel: @IdClass..... | 255 |
| 7.9.16 | Primärschlüssel - @EmbeddedId | 256 |
| 7.9.17 | Beispiel: @EmbeddedId..... | 257 |
| 7.9.18 | @Table..... | 258 |
| 7.9.19 | @Temporal..... | 259 |
| 7.9.20 | Beispiel: @Temporal..... | 260 |
| 7.9.21 | @Enumerated | 261 |
| 7.9.22 | Beispiel 1: @Enumerated | 262 |
| 7.9.23 | Beispiel 2: @Enumerated | 263 |
| 7.9.24 | @Column | 264 |
| 7.9.25 | @Column Parameter | 265 |
| 7.9.26 | @Lob..... | 267 |
| 7.9.27 | Beispiel: @Lob | 268 |
| 7.9.28 | Multi-table Mapping - @SecondaryTable..... | 269 |
| 7.9.29 | Beispiel: Multi-table Mapping | 270 |
| 7.9.30 | Eingebettete Objekte | 271 |
| 7.9.31 | Beispiel: Eingebettete Objekte | 272 |
| 7.9.32 | Eingebettete Objekte - @AttributeOverride..... | 273 |
| 7.9.33 | Beispiel: @AttributeOverride | 274 |
| 7.9.34 | Callbacks bei Entities..... | 275 |
| 7.10 | Entity Hibernate Special | 276 |
| 7.10.1 | Hibernate-Erweiterung - @DynamicInsert/Update | 276 |
| 7.10.2 | Hibernate-Erweiterung – Immutable | 279 |
| 7.11 | Entity Relationships..... | 280 |
| 7.11.1 | Motivation | 280 |
| 7.11.2 | Navigierbarkeit..... | 281 |
| 7.11.3 | Kardinalität..... | 282 |
| 7.11.4 | @OneToOne | 283 |
| 7.11.5 | @OneToMany | 284 |
| 7.11.6 | Beispiel: @OneToMany | 285 |
| 7.11.7 | @ManyToMany | 286 |
| 7.12 | Abfragen - Grundlagen | 287 |
| 7.12.1 | Motivation – Abfragen | 287 |
| 7.12.2 | Abfragen..... | 288 |
| 7.12.3 | JPQL | 289 |
| 7.12.4 | Typen von Abfragen..... | 290 |
| 7.12.5 | Beispiel 1 - <i>NativeQuery</i> | 291 |
| 7.12.6 | Beispiel 2 – <i>Query</i> | 292 |
| 7.12.7 | Beispiel 3 - <i>Query</i> mit Parameter..... | 293 |
| 7.12.8 | Beispiel 4 – <i>NamedQuery</i> | 294 |
| 7.12.9 | Fazit JPQL..... | 295 |
| 7.13 | Arbeiten mit JPQL..... | 296 |
| 7.13.1 | Anwendungsfälle | 296 |
| 7.13.2 | Selektionen..... | 297 |
| 7.13.3 | Beispiel Abfragen..... | 298 |
| 7.13.4 | JPQL <i>Expressions</i> (Pfade)..... | 300 |

| | | |
|----------|--|------------|
| 7.13.5 | Selektionen (DISTINCT) | 304 |
| 7.13.6 | Multiple Iterationsvariablen..... | 305 |
| 7.13.7 | Multiple Iterationsvariablen - Join (INNER)..... | 306 |
| 7.13.8 | Multiple Iterationsvariablen - Join (OUTER)..... | 307 |
| 7.13.9 | N+1 Select Problem..... | 308 |
| 7.13.10 | N+1 Select lösen - Inner Join | 309 |
| 7.13.11 | N+1 Select Problem - Hibernate..... | 310 |
| 7.13.12 | Mengen einschränken..... | 311 |
| 7.13.13 | Paging..... | 312 |
| 7.14 | Was ist neu in JPA 2.0? | 313 |
| 7.14.1 | JPA 1.0 vs. JPA 2.0 | 313 |
| 7.14.2 | Caching in JPA 1.x..... | 314 |
| 7.14.3 | Caching in JPA 2.x..... | 315 |
| 7.14.4 | Cache API - Second Level Cache | 316 |
| 7.14.5 | Neuigkeiten in JPA 2.0 – Mapping..... | 317 |
| 7.14.6 | Standard Properties in persistence.xml | 320 |
| 7.14.7 | Beispiel: Persistence.xml in JPA 1.0 | 321 |
| 7.14.8 | Beispiel: Persistence.xml in JPA 2.0 | 322 |
| 7.15 | Criteria API – Typesafe Queries | 323 |
| 7.15.1 | Motivation | 323 |
| 7.15.2 | Einführung in die Criteria-API..... | 324 |
| 7.15.3 | Motivation | 326 |
| 7.15.4 | JPA Criteria API..... | 327 |
| 7.15.5 | JPA Criteria API – Query-Komponenten..... | 328 |
| 7.15.6 | Begriffe..... | 329 |
| 7.15.7 | Expression<X> | 330 |
| 7.15.8 | Criteria Literale | 331 |
| 7.15.9 | Path<X> | 333 |
| 7.15.10 | Path<X> | 334 |
| 7.15.11 | Operatoren | 335 |
| 7.15.12 | Binäre Operatoren | 336 |
| 7.15.13 | Unäre Operatoren..... | 337 |
| 7.15.14 | String Manipulation | 338 |
| 7.15.15 | Date & Time Funktionen | 339 |
| 7.15.16 | Collections Manipulation | 340 |
| 7.15.17 | Vergleiche | 341 |
| 7.15.18 | Logische Operatoren | 342 |
| 7.15.19 | Criteria API – OO Abfragen..... | 343 |
| 7.15.20 | SELECT Klausel..... | 344 |
| 7.15.21 | SELECT DISTINCT | 345 |
| 7.15.22 | Multi SELECT | 346 |
| 7.15.23 | Multi SELECT – OO Alternative | 347 |
| 7.15.24 | CriteriaBuilder's construct | 348 |
| 7.15.25 | CriteriaBuilder Array..... | 349 |
| 7.15.26 | CriteriaBuilder TUPEL..... | 350 |
| 7.15.27 | Multi SELECT – Factory multiselect | 351 |
| 7.15.28 | JOIN Klausel..... | 352 |
| 7.15.29 | Criteria API – JOIN | 353 |
| 7.15.30 | Metamodell | 354 |
| 7.15.31 | Maven Konfiguration..... | 355 |
| 7.15.32 | Hibernate Metamodell Generator | 356 |
| 7.15.33 | Beispiel..... | 357 |
| 7.15.34 | Beispiel: Metamodell..... | 358 |
| 7.15.35 | Fazit | 359 |
| 8 | Java Persistenz - Vererbung..... | 361 |
| 8.1 | Einleitung..... | 362 |
| 8.2 | Strategien | 363 |
| 8.3 | Vererbungshierarchie | 364 |
| 8.4 | SINGLE_TABLE..... | 365 |

| | | |
|-----------|---|------------|
| 8.5 | TABLE_PER_CLASS..... | 370 |
| 8.6 | JOINED | 373 |
| 8.7 | Mapped Superclass..... | 376 |
| 8.8 | Transiente Klassen | 380 |
| 8.9 | Polymorphe Abfragen..... | 381 |
| 8.10 | Polymorphe Assoziationen | 384 |
| 8.11 | Fallstricke..... | 385 |
| 9 | Java Transaction Api (JTA) | 388 |
| 9.1 | Java Transaction API | 389 |
| 9.2 | Begriffe | 390 |
| 9.2.1 | Was ist eine Transaktion?..... | 390 |
| 9.2.2 | ACID-Eigenschaften | 391 |
| 9.2.3 | Isolation-Level | 392 |
| 9.2.4 | Grundlegende Probleme (ohne Transaktionen)..... | 393 |
| 9.2.5 | Transaktionen bei Konkurrenzgriff | 394 |
| 9.2.6 | Lösung der Probleme..... | 395 |
| 9.2.7 | Grenzen einer Transaktion..... | 396 |
| 9.2.8 | X/Open XA Protokoll (XA)..... | 397 |
| 9.2.9 | Verteilte (XA) Transaktionen | 398 |
| 9.3 | Java EE Transaktionen | 399 |
| 9.3.1 | JTA Package | 399 |
| 9.3.2 | Transaktionen API (JTA, JTS)..... | 400 |
| 9.3.3 | EJBs und Transaktionen | 401 |
| 9.3.4 | Container-Managed Transactions (CMT)..... | 402 |
| 9.3.5 | CMT – Transaktionsattribute | 403 |
| 9.3.6 | Transaktionsattribute Required | 404 |
| 9.3.7 | Transaktionsattribute Requires New..... | 405 |
| 9.3.8 | Weitere Transaktionsattribute..... | 406 |
| 9.3.9 | CMT – Transaktionsgrenzen | 407 |
| 9.3.10 | Bean-Managed Transactions (BMT)..... | 408 |
| 9.3.11 | (BMT) – UserTransaction Interface | 409 |
| 9.3.12 | BMT – Programmbeispiel..... | 410 |
| 9.3.13 | Fazit | 411 |
| 10 | Java Server Faces 2.0..... | 412 |
| 10.1 | MVC Model2 + Navigation..... | 413 |
| 10.1.1 | Inhalt | 413 |
| 10.1.2 | View | 414 |
| 10.1.3 | View Beispiel –Ansicht..... | 415 |
| 10.1.4 | View –Beispiel | 416 |
| 10.1.5 | JSF-View-Elemente | 417 |
| 10.1.6 | Expression Language | 418 |
| 10.1.7 | Model -Konfiguration über Annotations..... | 419 |
| 10.1.8 | Action-Methode..... | 420 |
| 10.1.9 | Implizite Navigation..... | 421 |
| 10.2 | Benutzeroberfläche | 422 |
| 10.2.1 | Inhalt | 422 |
| 10.2.2 | Auswahl einer View-Komponente..... | 423 |
| 10.2.3 | Basiselemente | 424 |
| 10.2.4 | Anzeigen von Listen..... | 425 |
| 10.2.5 | Komponente dataTable..... | 426 |
| 10.2.6 | Auswahl eines Elementes | 427 |
| 10.2.7 | Eigenschaften der Auswahl-Elemente..... | 428 |
| 10.2.8 | View zur Erzeugung von Auswahl-Elementen | 429 |
| 10.2.9 | Model für die Personenauswahl | 430 |
| 10.2.10 | Auswahl mehrerer Elemente | 431 |
| 10.2.11 | Multiselect – Eigenschaften..... | 432 |
| 10.2.12 | Model für die Mehrfachauswahl..... | 433 |
| 10.3 | Converter..... | 434 |

| | | |
|-----------|---|------------|
| 10.3.1 | Inhalt | 434 |
| 10.3.2 | Was sind Converter? | 435 |
| 10.3.3 | Standard-Converter | 436 |
| 10.3.4 | IntegerConverter..... | 437 |
| 10.3.5 | Number-Converter | 438 |
| 10.3.6 | Optionen zum Number-Converter..... | 439 |
| 10.3.7 | Converter erstellen..... | 440 |
| 10.3.8 | Konvertierung programmieren - im Konstruktor | 441 |
| 10.3.9 | Konvertierung programmieren - toString() | 442 |
| 10.3.10 | JSF-Converter | 443 |
| 10.4 | Validierung..... | 444 |
| 10.4.1 | Inhalt | 444 |
| 10.4.2 | Was ist Validierung? | 445 |
| 10.4.3 | Welche Validatoren gibt es?..... | 446 |
| 10.4.4 | Required..... | 447 |
| 10.4.5 | validateLength | 448 |
| 10.4.6 | Validator programmieren..... | 449 |
| 10.4.7 | Validator-Klasse..... | 450 |
| 10.4.8 | Validator einsetzen | 451 |
| 10.4.9 | Bean Validation..... | 452 |
| 11 | Sicherheit | 454 |
| 11.1 | Motivation - Sichere Java EE Applikation..... | 455 |
| 11.2 | Sicherheit – Definition | 456 |
| 11.3 | Sicherheit..... | 457 |
| 11.3.1 | Sicherheit im Enterprise Umfeld..... | 457 |
| 11.3.2 | Sichere Java EE Applikation? - Fallbeispiel 1 | 458 |
| 11.3.3 | Sichere Java EE Applikation? - Fallbeispiel 2..... | 459 |
| 11.3.4 | Sicherheitsrisiken (Top 10 Angriffsmuster) | 460 |
| 11.3.5 | (Standard-) Security Patterns | 461 |
| 11.4 | Java Sicherheit Grundlagen | 462 |
| 11.4.1 | JAAS - Java Sicherheit API..... | 462 |
| 11.4.2 | Authentication..... | 463 |
| 11.4.3 | JAAS – Authentifizierungsprozess..... | 464 |
| 11.4.4 | JAAS – LoginContext..... | 465 |
| 11.4.5 | JAAS – LoginModule | 466 |
| 11.4.6 | JAAS – CallbackHandler..... | 467 |
| 11.4.7 | Beispiel: Custom CallbackHandler | 468 |
| 11.4.8 | JAAS – Principal | 469 |
| 11.4.9 | JAAS – Configuration..... | 470 |
| 11.4.10 | JAAS - Config-file (Datei)..... | 471 |
| 11.4.11 | Beispiel: JAAS Authentifizierung | 472 |
| 11.4.12 | JAAS - Security-Manager..... | 473 |
| 11.4.13 | Autorization (Authorisation) | 474 |
| 11.4.14 | JAAS – Autorisierung..... | 475 |
| 11.4.15 | JAAS - Java User Permission | 476 |
| 11.4.16 | JAAS – Zugriffsrechte | 477 |
| 11.4.17 | JAAS – AccessController | 478 |
| 11.4.18 | JAAS - Autorisierung aktivieren..... | 480 |
| 11.4.19 | JAAS - Rollenbasierte Autorisierung | 481 |
| 11.4.20 | JACC - Autorisierung in der Servlet Spec..... | 482 |
| 11.4.21 | JAAC - Autorisierung in der EJB Spec..... | 483 |
| 11.4.22 | Beispiel 1: Security mit Apache Tomcat | 484 |
| 11.4.23 | Beispiel 2: Autorisierung in Wildfly..... | 485 |
| 11.4.24 | Fazit: (Web-) Sicherheit in Java EE..... | 486 |
| 11.5 | Sicherheit im Web..... | 487 |
| 11.5.1 | Security Constraint (Servlet 3.x)..... | 487 |
| 11.5.2 | Security Constraint (Servlet 2.x)..... | 488 |
| 11.5.3 | Security Constraint..... | 489 |
| 11.5.4 | Authentifizierung Verfahren..... | 490 |

- 11.5.5 Beispiel: FORM-Based Authentication.....491
- 11.5.6 FORM Authentication.....492
- 11.5.7 REST Sicherheit über web.xml.....493
- 11.5.8 Beispiel.....494
- 11.5.9 REST Sicherheit über Annotationen.....495
- 11.5.10 RolesAllowedDynamicFeature496
- 11.5.11 REST Sicherheit - HTTP Client497
- 11.6 Sicherheit Frameworks (Zeitgemäß).....498
 - 11.6.1 Frameworks.....498
 - 11.6.2 Oauth499
 - 11.6.3 OAuth-Protokoll500
 - 11.6.4 OAuth2.0.....501
 - 11.6.5 Was ist OAuth2?.....502
 - 11.6.6 OpenID Connect.....503
 - 11.6.7 KeyCloak.....504
- 12 Batch Processing.....505**
 - 12.1 Motivation506
 - 12.2 Stapelverarbeitung (Batch).....507
 - 12.3 Batch Applikationen.....508
 - 12.4 Batch Processing (JSR 352).....509
 - 12.4.1 Batch Architektur509
 - 12.4.2 JSR 352 - Batch Processing510
 - 12.4.3 JSR 352 - Java Batch API.....511
 - 12.4.4 JSR 352 - Java Batch API.....512
 - 12.5 Batch Artefakte mit Java EE 7 (JSR 352)513
 - 12.5.1 Batch Framework in Java EE 7513
 - 12.5.2 Infrastruktur-Komponenten.....514
 - 12.5.3 Struktur einer Batch Applikation (Job)515
 - 12.5.4 Job - Konfiguration & Packaging516
 - 12.5.5 Batch-Verarbeitung.....517
 - 12.5.6 Batchlet518
 - 12.5.7 Beispiel: Batchlet519
 - 12.5.8 *Chunk* (Block)520
 - 12.5.9 Standard Step Verarbeitung.....521
 - 12.5.10 Batch Checkpoints.....522
 - 12.5.11 Checkpoint-Policy (item)523
 - 12.5.12 Checkpoint-Policy (custom).....524
 - 12.5.13 Checkpoint Algorithmus525
 - 12.5.14 Chunk Verarbeitung mit Checkpointing526
 - 12.5.15 Batch Konstrukt527
 - 12.5.16 Chunk – ItemReader.....528
 - 12.5.17 Chunk – ItemProcessor.....529
 - 12.5.18 Chunk – ItemWriter.....530
 - 12.5.19 Parallelisierung - Partition-Map-Reduce531
 - 12.5.20 Partition-Map-Reduce532
 - 12.5.21 Partitionierung - statische Aufteilung533
 - 12.5.22 Partitionierung - dynamische Aufteilung534
 - 12.5.23 Parallelisierung durch Splits und Flows535
 - 12.5.24 Properties und Parameter536
 - 12.5.25 @BatchProperty537
 - 12.5.26 Job XML – Attribut-Substitution538
 - 12.5.27 Job-Metadaten.....539
 - 12.5.28 Batch Job starten und Status abfragen.....540
 - 12.5.29 Batch Job – Abfragen541
 - 12.5.30 Batch Job Status.....542
 - 12.5.31 Exceptions und Fehlerbehandlung543
 - 12.5.32 Exceptions abfangen544
 - 12.5.33 Injektion von JobContext, StepContext.....545
 - 12.5.34 Listener.....546

| | | |
|-----------|---|------------|
| 13 | Anhang: JAX-RS 1.0 Webservices | 547 |
| 13.1 | REST | 548 |
| 13.2 | RESTful Services mit Java EE..... | 549 |
| 13.3 | Ressourcen..... | 550 |
| 13.4 | JAX-RS Application..... | 551 |
| 13.5 | Jersey Servlet in web.xml..... | 552 |
| 13.6 | Application Class..... | 553 |
| 13.6.1 | Beispiel..... | 554 |
| 13.7 | Wichtige Annotationen | 555 |
| 13.7.1 | @Path | 556 |
| 13.7.2 | @Produces..... | 557 |
| 13.7.3 | @Consumes..... | 558 |
| 13.8 | Transformation von Daten | 559 |
| 13.9 | HTTP Response Codes..... | 560 |
| 13.10 | Services und Java-EE-Infrastruktur | 561 |
| 13.11 | JAX-RS mit EJB-Unterstützung | 562 |