

Java Programmierung Aufbau

Seminarunterlagen



einfach. gut. geschult.

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.

Adressen der ORDIX AG

Die ORDIX AG besitzt folgende Geschäftsstellen

ORDIX AG
Westernmauer 12-16
D-33098 Paderborn
Tel.: (+49) 0 52 51 / 10 63 - 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
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
Wikingerstr. 18-20
D-51107 Köln
Tel.: (+49) 02 21 / 8 70 61 - 0
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG
Marlene-Dietrich-Str. 5
D-89231 Neu-Ulm
Tel.: (+49) 07 31 / 9 85 88 -
Fax.: (+49) 01 80 / 1 67 34 90

Sie können die ORDIX AG von der ganzen Welt aus durch folgende Internet Adresse kontaktieren:

<http://www.ordix.de>

Sie können uns weiterhin über die Email-Adressen
training@ordix.de oder **info@ordix.de** kontaktieren

Inhaltsverzeichnis

1 Collection Framework	8
1.1 Was ist eine Collection?	8
1.2 Architektur des Collection Frameworks	8
1.3 Collection Framework Interfaces	10
1.3.1 Interface Collection	10
1.3.2 Interface List	10
1.3.3 Interface Set	10
1.3.4 Interface SortedSet	12
1.3.5 Interface Map	12
1.3.6 Interface SortedMap	12
1.4 LinkedList	14
1.5 ArrayList	16
1.6 HashSet	18
1.7 TreeSet	20
1.8 HashMap	22
1.9 TreeMap	24
1.10 Interface Iterator	26
1.11 Interface ListIterator	28
1.12 Klasse Collections	30
1.13 Algorithmen - Sortieren	30
1.14 Algorithmus - Permutieren	32
1.15 Algorithmus - Minimum & Maximum	32
1.16 Algorithmus - Suchen	34
1.17 Algorithmus - Datenmanipulation	34
1.18 Synchronisierte Wrapper	36
1.19 Nicht modifizierbare Wrapper	38
1.20 Zusätzliche Funktionalität	40
1.21 Einbindung bestehender Collections	42
1.22 Aufwärts - Kompatibilität	44
1.23 Abwärts - Kompatibilität	46
2 Generics	50
2.1 Java Generics	50
2.2 Collections bisher	52
2.3 Generische Collections	52
2.4 Generische Collections mit mehreren Typparametern	54
2.5 Ober und Unterklassen in generischen Typen	56
2.6 weak und strong typing	58
2.7 Die „andere Seite“	60
2.8 Eigene generische Typen	62
2.9 TreeMap ohne Bounds	66
2.10 Bounds	70
2.11 TreeMap mit Bounds	72
2.12 Generische Methoden	74
2.12.1 generische Methoden Java 1.4	76
2.12.2 generische Methoden Java 1.5	78
2.13 Typ-Inferenz	80

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

2.14 Wildcard Instanziierung	82
2.15 Wildcard Beispiele	84
2.16 Type Erasure	86
2.17 Generische Typen - Grenzen	88
3 Objekt Serialisierung	92
3.1 Serialisierung eines Objektes:	94
3.2 Deserialisierung eines Objektes:	96
3.3 Voraussetzungen zur Objekt Serialisierung.....	98
3.4 Dateiformat der Objektserialisierung	100
3.5 Problem der Speicherung von Objektreferenzen.....	104
3.6 Prinzip der Speicherung von Objektreferenzen	106
3.7 Sicherheit.....	108
3.8 Erweiterung der Serialisierung und Deserialisierung.....	110
3.9 Interface Externalizable	112
3.10 Sicherheitstipps	114
3.11 Versionsverwaltung	116
3.12 Serialisierung zum Klonen	118
4 Innere Klassen	122
4.1 Was sind Innere Klassen	122
4.2 "Normale" Toplevel Klassen	124
4.3 Beispiel für Innere Klassen	126
4.4 Typisierung von Inneren Klassen	128
4.5 Statische innere Klassen	130
4.6 Nicht-statische innere Klassen	132
4.7 Member Klassen.....	134
4.8 Beispiel: Member Klassen	136
4.9 Lokale Klassen	138
4.10 Beispiel: Lokale Klassen.....	140
4.11 Anonyme Klassen.....	142
4.12 Beispiel: Anonyme Klassen	144
4.13 Anwendung: Implementation Hiding.....	146
4.14 Anwendung: Adapter Klassen	148
5 Typesafe Enums	152
5.1 Aufzählungen bisher	152
5.2 Probleme mit int-Enumerations	154
5.3 Idiom für typsichere Aufzählungen	156
5.4 Typesafe Enums.....	160
5.5 Deklaration von enum-Klassen.....	162
5.6 Eigenschaften von enum-Klassen	164
5.7 enum Beispiel	166
5.8 Methoden in enum-Klassen.....	170
5.8.1 switch-Statement	172
5.8.2 abstrakte Methoden.....	172
5.9 Zwei neue Klassen für enum	174
5.9.1 EnumSet.....	176
5.9.2 EnumMap	178
6 Reflection API.....	182
6.1 Reflection API	182
6.2 Klassen der Reflection API	184

6.3 Klassen erkunden	186
6.4 Klasse ermitteln	188
6.5 Klassennamen ermitteln	188
6.6 Klassenmodifizierer ermitteln	190
6.7 Superklasse ermitteln	192
6.8 Interface einer Klasse ermitteln	192
6.9 Interface-Eigenschaft abprüfen	194
6.10 Klassen bzw. Interfaceattribute ermitteln	194
6.11 Konstruktoren ermitteln	196
6.12 Methoden samt Signatur ermitteln	196
6.13 Objekte manipulieren	198
6.14 Objekte über den Default-Konstruktor erzeugen	198
6.15 Objekte über parametrisierten Konstruktor erzeugen	200
6.16 Attributwerte ermitteln	202
6.17 Attributwerte setzen	204
6.18 Methoden aufrufen	206
6.19 Arbeiten mit Arrays	208
6.20 Array-Eigenschaft abprüfen	210
7 Java Database Connectivity (JDBC)	214
7.1 Zugriff auf Datenbanken in der Nicht-Java-Welt	214
7.2 Behandlung verschiedener Datenbanken	214
7.3 Open Database Connectivity (ODBC)	216
7.4 Java Database Connectivity (JDBC)	216
7.5 Eigenschaften des JDBC	218
7.6 Beziehung zwischen JDBC-Klassen	220
7.7 DB-Verbindung aufbauen	222
7.8 SQL-Anweisungen absetzen	224
7.9 Interface Statement	224
7.10 Beispiel: SQL-Anweisungen absetzen	226
7.11 Auslesen einer Ergebnismenge	228
7.12 Interface ResultSet	228
7.13 Beispiel: Auslesen einer Ergebnismenge	230
7.14 Zuordnung JDBC Typ zu Java Typ	232
7.15 Schließen einer DB-Verbindung	232
7.16 Vorbereitete Anweisungen	234
7.17 JDBC und Transaktionen	236
7.18 Transaktions-Modi	238
7.19 Beispiel mit Transaktionen	240
7.20 JDBC und Metadaten	242
7.21 Interface DatabaseMetaData	242
7.22 Interface ResultSetMetaData	244
8 Annotations	248
8.1 Was sind Annotations ?	248
8.2 Anwendungsgebiete	250
8.3 Annotation Beispiel: @Deprecated	252
8.4 Verwendung von Annotations	254
8.5 Packages für Annotations	256
8.6 Vordefinierte Annotations	258
8.7 Standard Annotations	260
8.8 Beispiel: @SupressWarnings	262
8.9 Definition eigener Annotations	264

8.10 Annotations mit Eigenschaften	266
8.11 Meta-Annotation Target	270
8.12 Beispiel: @Target	272
8.13 Meta-Annotation Retention	274
8.14 Beispiel: @Retention	276
8.15 Meta-Annotation Documented	278
8.16 Beispiel: @Documented	280
8.17 Meta-Annotation Inherited	282
8.18 Beispiel: @Inherited.....	282
8.19 Zugriff über Reflection	284
8.20 Interface AnnotatedElement	284
8.21 Vorgehensweise mit Reflection	286
8.22 Beispiel: Zugriff über Reflection.....	288
9 Garbage Collection	292
9.1 Was ist die Garbage Collection (GC).....	292
9.2 finalize()-Methode	294
9.3 „Teilweise“ und „Vollständige“ GC	296
9.4 Unterschiedliche Parametereinstellungen	298
9.5 Anpassung der initialen und max. Speichergröße	300
9.6 Verfolgen der Garbage Collection Aktivität.....	302
9.7 „Generational Virtual Machine“	304
9.8 Anpassen der „nursery“-Speichergröße	306
9.9 „Incremental Mode“.....	308