

Java Programmierung Grundlagen

Seminarunterlage

Version: 6.01

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
Edmund-Rumpler-Straße 7
D-51149 Köln
Tel.: (+49) 02 21 / 8 70 61 – 0
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG
Südwestpark 67/2
D-90449 Nürnberg
Tel.: (+49) 0 52 51 / 10 63 - 0
Fax.: (+49) 01 80 / 1 67 34 90

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

Email: seminare@ordix.de

Inhaltsverzeichnis

1	Grundlagen der Sprache	8
1.1	Was ist Java?	9
1.2	Java-Design Kriterien	10
1.2.1	Einfach und objektorientiert	10
1.2.2	Verteilt und interpretiert	12
1.2.3	Robust und sicher	13
1.2.4	Architekturneutral und portabel	14
1.2.5	Hochleistungsfähig und multithreaded	15
1.2.6	Dynamisch	16
1.3	Historie	17
1.4	Anwendungsgebiete	19
1.5	Java Komponenten	20
1.5.1	Java Virtual Machine (JVM)	21
1.5.2	Was macht die JVM	22
1.5.3	Java Runtime Environment (JRE)	23
1.5.4	Java Development Kit (JDK)	24
1.6	.java und *.class-Datei	25
1.7	Java-Anwendungen	26
1.7.1	Einfache Java-Anwendung	27
1.7.2	Kompilieren und Starten einer Java-Anwendung	28
1.8	Integrated Development Environment	29
1.8.1	Erstellen eines Projektes mit Eclipse	30
1.9	Die Main Methode	31
2	Einführung in die Objektorientierung.....	32
2.1	Wieso Objektorientierung?	33
2.2	Abstraktion	34
2.3	Prozedurale vs. objektorientierte Programmierung	35
2.4	Unified Modeling Language	36
2.4.1	Struktur- und Verhaltensdiagramme	37
2.4.2	Klassendiagramm und Klasse	38
2.5	Objekt	39
2.5.1	Attribut	40
2.5.2	Operation, Methode	41
2.6	Schnittstelle, Schnittstellenklasse	42
2.7	Entwurfsmuster	43
2.8	Paket	44
2.9	Klassendiagramm (Beziehungselemente)	45
2.9.1	Generalisierung, Spezialisierung	46
2.9.2	Beispiel: Generalisierung, Spezialisierung	47
2.10	Assoziation	48
2.10.1	Gerichtete Assoziation	49
2.11	Aggregation und Komposition	50
2.12	Verhaltensdiagramme	51
2.12.1	Sequenzdiagramm	52
3	Klassen und Objekte	53
3.1	Objektorientierung in Java	54
3.2	Klassen	55
3.3	Attribute und Methoden	57
3.4	Sichtbarkeitsbereiche	58
3.4.1	Sichtbarkeitsbereich – Public und protected	59
3.4.2	Sichtbarkeitsbereich – Package und private	60
3.4.3	Sichtbarkeitsbereich – Zusammenfassung	61
3.5	Syntax einer Klassendefinition	62
3.6	Von Klassen zu Objekten	63
3.7	new-Operator	64
3.8	Konstruktor	65

3.8.1	Definition von Konstruktoren	66
3.9	Zugriff auf Datentypen.....	68
3.10	Equals und Hashcode	72
3.11	this-Referenz	73
3.12	Instanzvariablen	75
3.13	Instanzmethoden.....	76
3.14	Eigenschaften von Methoden.....	77
3.15	Beispiel Parameterübergabe.....	79
3.16	return-Anweisung	80
3.17	static: Klassenvariablen und -methoden	81
3.18	Beispiel: static	82
3.19	final-Attribute	83
3.20	Exkurs: Parameter als varargs.....	84
3.21	Zusammenfassung.....	85
4	Array, String, StringBuilder, Wrapper, Auto(un)boxing	86
4.1	Arrays	87
4.2	Erzeugen von Array-Objekten.....	89
4.2.1	Arrays einfacher Werte.....	90
4.3	Multidimensionale Arrays	91
4.3.1	Bemerkung zu multidimensionalen Arrays.....	92
4.4	Weiteres zu Arrays.....	93
4.5	Klasse String	94
4.5.1	Einige String-Methoden.....	95
4.5.2	Konkatenationsoperator für Strings.....	96
4.5.3	Klasse StringBuilder	97
4.5.4	Einige StringBuilder-Methoden	98
4.6	Wrapper-Klassen für primitive Typen.....	99
4.6.1	Beispiele mit der Klasse Integer.....	100
4.7	Eigenschaften von Float und Double	102
4.8	Boxing und Unboxing	103
4.8.1	Manuelles vs. automatisches Boxing	104
4.9	Autoboxing und Arrays speziell.....	106
4.10	Nutzung von Auto(un)boxing.....	108
4.11	Achtung: Referenzgleichheit	109
4.12	Zusammenfassung.....	111
5	Ablaufsteuerung: if, switch, while, do...while, for, for each	112
5.1	Ablaufsteuerung in Java.....	113
5.2	Vergleichsoperatoren	114
5.3	Logische Operatoren.....	115
5.4	Bitweise Operatoren.....	116
5.5	if-Anweisung.....	117
5.6	Verschachtelte if-Anweisungen.....	118
5.7	Alternative zur if-Anweisung.....	119
5.8	switch-Anweisung	120
5.9	while-Schleife	122
5.10	do...while-Schleife	123
5.11	for-Schleife	124
5.12	for each-Schleife	126
5.13	Geschachtelte Schleifen	128
5.14	Verwendung von Enhanced for Loop.....	131
5.15	break-Anweisung	133
5.16	continue-Anweisung	135
5.17	Zusammenfassung.....	137
6	Vererbung.....	138
6.1	Vererbung.....	139
6.2	Überschreiben von Methoden	142
6.3	Beispiel: Überschreiben	143

6.4	super-Zugriff auf Superklasse	144
6.5	Beispiel: super	145
6.6	Abstrakte Klassen	147
6.7	Abstrakte Methoden	148
6.8	Abstrakte Klassen: Vererbung	149
6.9	Wozu abstrakte Klassen	151
6.10	Beispiel	152
6.11	Verwendung als Typ	153
6.12	Polymorphie	154
6.12.1	Polymorphie: Beispiel mit abstrakten Klassen	155
6.13	Interfaces	156
6.13.1	Weiteres zu Interfaces	157
6.13.2	Interfaces implementieren	158
6.13.3	Methoden überschreiben	159
6.13.4	Interfaces als Typ	161
6.13.5	Interfaces als Typ, Beispiel	162
6.13.6	Anwendungsbeispiel	163
6.14	Zusammenfassung	168
7	Collection Framework	169
7.1	Was ist eine Collection?	170
7.2	Architektur des Collection Frameworks	171
7.3	Collection Framework Interfaces	172
7.4	Ein kurzer Exkurs zu Generics	173
7.5	Interface Iterable	174
7.6	Interface Iterator	175
7.7	Interface List	176
7.8	Array List	177
7.9	LinkedList	178
7.10	Unmodifiable List	179
7.11	Interface Set	180
7.12	HashSet	181
7.13	TreeSet	182
7.14	Unmodifiable Set	183
7.15	Interface Map	184
7.16	HashMap	185
7.17	TreeMap	186
7.18	Klasse Collections	187
7.18.1	Algorithmen – Sortieren	188
7.18.2	Algorithmen – Permutieren	189
7.18.3	Algorithmen – Minimum & Maximum	190
7.18.4	Algorithmen – Suchen	191
7.18.5	Algorithmen – Datenmanipulation	192
7.18.6	Nicht modifizierbare Wrapper	193
7.18.7	Zusätzliche Funktionalität	194
8	Lambdas und Stream API	195
8.1	Warum gibt es Lambda-Ausdrücke?	196
8.2	Lambda-Ausdruck am Beispiel Comparator	197
8.3	Syntax eines Lambda-Ausdruckes	198
8.4	Inline-Syntax und Methodenreferenzen	199
8.5	Stream API	200
8.6	Das Stream-Interface	201
8.7	Stream API – Beispiel	202
9	Ausnahmebehandlung – Exception Handling	203
9.1	Ausnahme (Exception)	204
9.2	Exception-Klassen	205
9.3	Vererbungshierarchie	206
9.4	Exception Handling	208

9.5	Exception Handling – Java 7	212
9.6	finally-Block	213
9.7	Auslösen von Ausnahmen	214
9.8	PrintStackTrace	215
9.9	Auswerten von Ausnahmen	216
9.10	Definition eigener Ausnahmen	218
9.11	Tipps	219
9.12	Zusammenfassung	221
10	Pakete	222
10.1	Paket	223
10.2	Pakete	224
10.3	Erstellung eigener Pakete	225
10.4	Paketstruktur	226
10.5	Einbinden von Paketen	227
10.6	Finden von Paketen	228
10.7	Static Import	229
11	Design Pattern	230
11.1	Design Pattern	231
11.2	Design Pattern – Übersicht	232
11.3	Erzeugungsmuster	233
11.4	Erzeugungsmuster – Singleton	234
11.5	Strukturmuster	236
11.6	Strukturmuster – Adaptor	237
11.7	Verhaltensmuster	238
11.8	Verhaltensmuster – Beobachter	239
12	Überblick API	241
12.1	Java und I/O – Überblick	242
12.2	Historie – Warum wurde java.io überarbeitet	243
12.3	Neues IO (NIO)	244
12.4	Buffer	245
12.5	Channel	246
12.6	NIO 2	247
12.7	Path – Alles hat einen Pfad	248
12.7.1	Path – Relativ oder absolut	249
12.7.2	Path – Nützliche Methoden	250
12.8	Files – Mit Dateien arbeiten	251
12.8.1	Files – Methoden	252
12.9	Anlegen & Schreiben einer Datei	253
12.9.1	Dateien schreiben	254
12.10	Lesen einer Datei	255
12.10.1	Dateien Lesen	256
12.11	NIO Zusammenfassung	257
12.12	Date/Time API (Java 8)	258
12.13	LocalDate, LocalTime und LocalDateTime	259
12.14	ZonedDateTime, ZonedTime und ZonedDateTime	260
12.15	Arbeiten mit der Date API	261
12.16	DateTimeFormatter	263
13	Git	264
13.1	Was ist Git?	265
13.2	Git Zustände	266
13.3	Git clone	267
13.4	Git add	268
13.5	Git commit	269
13.6	Git pull	270
13.7	Git push	271
13.8	Weitere nützliche Befehle	272

13.9	Tag- und Branch-Konzept	273
13.10	EGit Plugin für Eclipse	274
13.11	Eclipse – Konfiguration Git (Properties)	275
13.12	Eclipse – Konfiguration Git (Repository)	276
14	JUnit.....	277
14.1	Testen mit JUnit	278
14.2	Unterschied Unit Test und Integrationstest.....	279
14.3	Strukturierung von Unit Tests.....	280
14.4	JUnit 4	281
14.5	JUnit 5 – „JUnit Lambda“	283
15	Bibliotheken/Frameworks.....	284
15.1	Frameworks.....	285
15.2	Java EE?	286
15.3	Aufgaben eines Java EE Servers	287
15.4	Java Persistence API (JPA)	288
15.5	ORM – Objektrationales Mapping	289
15.6	Spring	290
15.7	Warum Spring?	291
15.8	JavaServer Faces (JSF)	292
15.9	Aufgaben von JSF	293
15.10	JAXB	294
15.11	XML Data Binding, Primäre Ziele.....	295