

# ***Java Programmierung Grundlagen***

## ***Seminarunterlage***

***Version: 5.03***



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](mailto:seminare@ordix.de)

# Inhaltsverzeichnis

<b>1</b>	<b>Grundlagen der Sprache</b>	<b>8</b>
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	20
1.5	Java Komponenten	21
1.5.1	Java Virtual Machine (JVM)	22
1.5.2	Was macht die JVM	23
1.5.3	Java Runtime Environment (JRE)	24
1.5.4	Java Development Kit (JDK)	25
1.6	.java und *.class-Datei	26
1.7	Java-Anwendungen	27
1.7.1	Einfache Java-Anwendung	28
1.7.2	Kompilieren und Starten einer Java-Anwendung	29
1.8	Integrated Development Environment	30
1.8.1	Erstellen eines Projektes mit Eclipse	31
1.9	Die Main Methode	32
<b>2</b>	<b>Einführung in die Objektorientierung</b>	<b>33</b>
2.1	Wieso Objektorientierung?	34
2.2	Abstraktion	35
2.3	Prozedurale vs. Objektorientierte Programmierung	36
2.4	Unified Modeling Language	37
2.4.1	Struktur- und Verhaltensdiagramme	38
2.4.2	Klassendiagramm und Klasse	39
2.5	Objekt	40
2.5.1	Attribut	41
2.5.2	Operation, Methode	42
2.6	Schnittstelle, Schnittstellenklasse	43
2.7	Entwurfsmuster	44
2.8	Paket	45
2.9	Klassendiagramm (Beziehungselemente)	46
2.9.1	Generalisierung, Spezialisierung	47
2.9.2	Beispiel: Generalisierung, Spezialisierung	48
2.10	Assoziation	49
2.10.1	Gerichtete Assoziation	50
2.11	Aggregation und Komposition	51
2.12	Verhaltensdiagramme	52
2.12.1	Sequenzdiagramm	53
<b>3</b>	<b>Klassen und Objekte</b>	<b>54</b>
3.1	Objektorientierung in Java	55
3.2	Klassen	56
3.3	Attribute und Methoden	58
3.4	Sichtbarkeitsbereiche	59
3.4.1	Sichtbarkeitsbereich – Public und Protected	60
3.4.2	Sichtbarkeitsbereich – Package und Private	61
3.4.3	Sichtbarkeitsbereich - Zusammenfassung	62
3.5	Syntax einer Klassendefinition	63
3.6	Von Klassen zu Objekten	64
3.7	new-Operator	65
3.8	Konstruktor	66

3.8.1	Definition von Konstruktoren .....	67
3.9	Zugriff auf Datentypen.....	69
3.10	Equals und Hashcode .....	73
3.11	this-Referenz .....	74
3.12	Instanzvariablen .....	76
3.13	Instanzmethoden.....	77
3.14	Eigenschaften von Methoden .....	78
3.15	Beispiel Parameterübergabe.....	80
3.16	return-Anweisung .....	81
3.17	static: Klassenvariablen und -methoden .....	82
3.18	Beispiel: static .....	83
3.19	final-Attribute .....	84
3.20	Exkurs: Parameter als varargs.....	85
3.21	Zusammenfassung.....	86
<b>4</b>	<b>Array, String, StringBuilder, Wrapper, Auto(un)boxing .....</b>	<b>87</b>
4.1	Arrays .....	88
4.2	Erzeugen von Array-Objekten.....	90
4.2.1	Arrays einfacher Werte.....	91
4.3	Multidimensionale Arrays .....	92
4.3.1	Bemerkung zu multidimensionalen Arrays.....	93
4.4	Weiteres zu Arrays .....	94
4.5	Klasse String .....	95
4.5.1	Einige String-Methoden.....	96
4.5.2	Konkatenationsoperator für Strings.....	97
4.5.3	Klasse StringBuilder .....	98
4.5.4	Einige StringBuilder-Methoden .....	99
4.6	Wrapper-Klassen für primitive Typen.....	100
4.6.1	Beispiele mit der Klasse Integer.....	101
4.7	Eigenschaften von Float und Double .....	103
4.8	Boxing und Unboxing .....	104
4.8.1	Manuelles vs. automatisches Boxing .....	105
4.9	Autoboxing und Arrays speziell.....	107
4.10	Nutzung von Auto(un)boxing.....	109
4.11	Achtung: Referenzgleichheit .....	110
4.12	Zusammenfassung.....	112
<b>5</b>	<b>Ablaufsteuerung: if, switch, while, do..while, for, for each .....</b>	<b>113</b>
5.1	Ablaufsteuerung in Java.....	114
5.2	Vergleichsoperatoren .....	115
5.3	Logische Operatoren.....	116
5.4	Bitweise Operatoren.....	117
5.5	if-Anweisung.....	118
5.6	Verschachtelte if-Anweisungen.....	119
5.7	Alternative zur if-Anweisung.....	120
5.8	switch-Anweisung .....	121
5.9	while-Schleife .....	123
5.10	do..while-Schleife .....	124
5.11	for-Schleife .....	125
5.12	for each-Schleife .....	127
5.13	Geschachtelte Schleifen .....	129
5.14	Verwendung von Enhanced for Loop.....	132
5.15	break-Anweisung .....	134
5.16	continue-Anweisung.....	136
5.17	Zusammenfassung.....	138
<b>6</b>	<b>Vererbung.....</b>	<b>139</b>
6.1	Vererbung.....	140
6.2	Überschreiben von Methoden .....	143
6.3	Beispiel: Überschreiben .....	144

6.4	super-Zugriff auf Superklasse .....	146
6.5	Beispiel: super .....	147
6.6	Abstrakte Klassen .....	149
6.7	Abstrakte Methoden .....	150
6.8	Abstrakte Klassen: Vererbung .....	151
6.9	Wozu abstrakte Klassen .....	153
6.10	Beispiel.....	154
6.11	Verwendung als Typ .....	155
6.12	Polymorphie .....	156
6.12.1	Polymorphie: Beispiel mit abstrakten Klassen .....	157
6.13	Interfaces.....	158
6.13.1	Weiteres zu Interfaces.....	159
6.13.2	Interfaces implementieren .....	160
6.13.3	Methoden überschreiben .....	161
6.13.4	Interfaces als Typ .....	163
6.13.5	Interfaces als Typ, Beispiel .....	164
6.13.6	Anwendungsbeispiel .....	165
6.14	Zusammenfassung.....	170
<b>7</b>	<b>Collection Framework.....</b>	<b>171</b>
7.1	Was ist eine Collection? .....	172
7.2	Architektur des Collection Frameworks .....	173
7.3	Collection Framework Interfaces .....	174
7.4	Einfaches Beispiel einer Collection .....	175
7.5	Interface Iterator .....	176
7.5.1	Interface Iterable .....	177
7.5.2	Interface ListIterator .....	178
7.6	LinkedList .....	179
7.7	ArrayList .....	181
7.8	HashSet.....	183
7.9	TreeSet.....	184
7.10	HashMap .....	185
7.11	TreeMap .....	187
7.12	Klasse Collections .....	189
7.12.1	Algorithmen – Sortieren.....	190
7.12.2	Algorithmen – Permutieren .....	191
7.12.3	Algorithmen - Minimum & Maximum .....	192
7.12.4	Algorithmen – Suchen .....	193
7.12.5	Algorithmen – Datenmanipulation .....	194
7.12.6	Synchronisierte Wrapper.....	195
7.12.7	Nicht modifizierbare Wrapper.....	196
7.12.8	Zusätzliche Funktionalität.....	197
<b>8</b>	<b>Ausnahmebehandlung – Exception Handling.....</b>	<b>198</b>
8.1	Ausnahme (Exception).....	199
8.2	Exception-Klassen .....	200
8.3	Vererbungshierarchie.....	201
8.4	Exception Handling .....	203
8.5	Exception Handling – Java 7.....	207
8.6	finally-Block .....	208
8.7	Auslösen von Ausnahmen .....	209
8.8	PrintStackTrace.....	210
8.9	Auswerten von Ausnahmen .....	211
8.10	Definition eigener Ausnahmen.....	213
8.11	Tipps.....	214
<b>9</b>	<b>Pakete .....</b>	<b>216</b>
9.1	Paket .....	217
9.2	Pakete .....	218
9.3	Erstellung eigener Pakete .....	219

9.4	Paketstruktur .....	220
9.5	Einbinden von Paketen .....	221
9.6	Finden von Paketen .....	222
9.7	Static Import .....	223
<b>10</b>	<b>Design Pattern .....</b>	<b>224</b>
10.1	Design Pattern .....	225
10.2	Design Pattern - Übersicht .....	226
10.3	Erzeugungsmuster .....	227
10.4	Erzeugungsmuster Singleton .....	228
10.5	Strukturmuster .....	230
10.6	Strukturmuster Adaptor .....	231
10.7	Verhaltensmuster .....	232
10.8	Verhaltensmuster - Beobachter .....	233
<b>11</b>	<b>Überblick API .....</b>	<b>235</b>
11.1	Java und I/O – Überblick .....	236
11.2	Historie – Warum wurde java.io überarbeitet .....	237
11.3	Neues IO (NIO) .....	238
11.4	Buffer .....	239
11.5	Channel .....	240
11.6	NIO 2 .....	241
11.7	Path – Alles hat einen Pfad .....	242
	11.7.1 Path – Relativ oder Absolut .....	243
	11.7.2 Path – Nützliche Methoden .....	244
11.8	Files – Mit Dateien arbeiten .....	245
	11.8.1 Files - Methoden .....	246
11.9	Anlegen & Schreiben einer Datei .....	247
	11.9.1 Dateien schreiben .....	248
11.10	Lesen einer Datei .....	249
	11.10.1 Dateien Lesen .....	250
11.11	NIO Zusammenfassung .....	251
11.12	Date/Time API (Java 8) .....	252
11.13	LocalDate, LocalTime und LocalDateTime .....	253
11.14	ZonedDateTime, ZonedTime und ZonedDateTime .....	254
11.15	Arbeiten mit der Date API .....	255
11.16	DateTimeFormatter .....	257
<b>12</b>	<b>GIT .....</b>	<b>258</b>
12.1	Was ist GIT? .....	259
12.2	Git Zustände .....	260
12.3	Git clone .....	261
12.4	Git Add .....	262
12.5	Git commit .....	263
12.6	Git pull .....	264
12.7	Git push .....	265
12.8	Weitere nützliche Befehle .....	266
12.9	Tag- und Branch- Konzept .....	267
12.10	EGit Plugin für Eclipse .....	268
12.11	Eclipse – Konfiguration Git (Properties) .....	269
12.12	Eclipse – Konfiguration Git (Repository) .....	270
<b>13</b>	<b>JUnit .....</b>	<b>271</b>
13.1	Testen mit JUnit .....	272
13.2	Unterschied Unit Test vs Integrationstest .....	273
13.3	Strukturierung von Unit Tests .....	274
13.4	JUnit 4 .....	275
13.5	Junit 5 – „JUnit Lambda“ .....	277
<b>14</b>	<b>Bibliotheken/Frameworks .....</b>	<b>278</b>

---

14.1	Frameworks .....	279
14.2	Java EE? .....	280
14.3	Aufgaben eines Java EE Servers .....	281
14.4	Java Persistence API (JPA) .....	282
14.5	ORM – Objektrationales Mapping .....	283
14.6	Spring .....	284
14.7	Warum Spring? .....	285
14.8	JavaServer Faces (JSF) .....	286
14.9	Aufgaben von JSF .....	287
14.10	JAXB .....	288
14.11	XML Data Binding, Primäre Ziele.....	289