

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

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.....	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
2	Einführung in die Objektorientierung	33
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
3	Klassen und Objekte.....	54
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
4	Array, String, StringBuilder, Wrapper, Auto(un)boxing	87
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
5	Ablaufsteuerung: if, switch, while, do...while, for, for each	113
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
6	Vererbung.....	139
6.1	Vererbung.....	140
6.2	Überschreiben von Methoden	143
6.3	Beispiel: Überschreiben	144

6.4	super-Zugriff auf Superklasse	145
6.5	Beispiel: super	146
6.6	Abstrakte Klassen	148
6.7	Abstrakte Methoden	149
6.8	Abstrakte Klassen: Vererbung	150
6.9	Wozu abstrakte Klassen	152
6.10	Beispiel.....	153
6.11	Verwendung als Typ	154
6.12	Polymorphie	155
6.12.1	Polymorphie: Beispiel mit abstrakten Klassen	156
6.13	Interfaces.....	157
6.13.1	Weiteres zu Interfaces.....	158
6.13.2	Interfaces implementieren	159
6.13.3	Methoden überschreiben	160
6.13.4	Interfaces als Typ	162
6.13.5	Interfaces als Typ, Beispiel	163
6.13.6	Anwendungsbeispiel	164
6.14	Zusammenfassung.....	169
7	Collection Framework.....	170
7.1	Was ist eine Collection?.....	171
7.2	Architektur des Collection Frameworks	172
7.3	Collection Framework Interfaces	173
7.4	Einfaches Beispiel einer Collection	174
7.5	Interface Iterator.....	175
7.5.1	Interface Iterable	176
7.5.2	Interface ListIterator	177
7.6	LinkedList	178
7.7	ArrayList	180
7.8	HashSet.....	182
7.9	TreeSet.....	183
7.10	HashMap	184
7.11	TreeMap.....	186
7.12	Klasse Collections.....	188
7.12.1	Algorithmen – Sortieren.....	189
7.12.2	Algorithmen – Permutieren	190
7.12.3	Algorithmen - Minimum & Maximum	191
7.12.4	Algorithmen – Suchen	192
7.12.5	Algorithmen – Datenmanipulation	193
7.12.6	Synchronisierte Wrapper.....	194
7.12.7	Nicht modifizierbare Wrapper.....	195
7.12.8	Zusätzliche Funktionalität.....	196
8	Ausnahmebehandlung – Exception Handling.....	197
8.1	Ausnahme (Exception).....	198
8.2	Exception-Klassen	199
8.3	Vererbungshierarchie.....	200
8.4	Exception Handling	202
8.5	Exception Handling – Java 7.....	206
8.6	finally-Block	207
8.7	Auslösen von Ausnahmen	208
8.8	PrintStackTrace.....	209
8.9	Auswerten von Ausnahmen	210
8.10	Definition eigener Ausnahmen.....	212
8.11	Tipps.....	213
9	Pakete.....	216
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
10	Design Pattern	224
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
11	Überblick API	235
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
12	GIT	258
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
13	JUnit	271
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
14	Bibliotheken/Frameworks	278

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