

Java Programmierung Grundlagen

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 Grundlagen der Sprache	10
1.1 Was ist Java?	10
1.2 Java-Design Kriterien	12
1.2.1 Einfach und objektorientiert	12
1.2.2 Verteilt und Interpretiert	14
1.2.3 Robust und Sicher	14
1.2.4 Architekturneutral und portabel	16
1.2.5 Hochleistungsfähig und Multithreaded	16
1.2.6 Dynamisch.....	18
1.3 Java Virtual Machine (JVM).....	20
1.4 Was macht die JVM?	20
1.5 Just-in-Time (JIT)-Compiler	22
1.6 .java und *.class-Datei	22
1.7 Java-Anwendungen	24
1.8 Java-Applets	24
1.9 Java Beans	26
1.10 Einfache Java-Anwendung	28
1.11 Kompilieren und Starten einer Java-Anwendung	28
1.12 Kommentare	30
1.13 Programmier-Tools	32
2 Variablen, Datentypen und Operatoren	36
2.1 Variablendeklaration und -initialisierung	36
2.2 Variablennamen	36
2.3 Reservierte Schlüsselwörter	38
2.4 Variablentypen in Java.....	40
2.5 Einfache (primitive) Datentypen	40
2.6 Angabe von Literalen.....	42
2.7 Anmerkungen zu Literalen.....	42
2.8 Arithmetische Operationen	44
2.9 Zuweisungsoperatoren	44
2.10 Umwandlungen und Casting	46
2.11 Inkrement- und Dekrementoperatoren.....	48
2.12 Vergleichsoperatoren.....	50
2.13 Logische Operatoren	50
2.14 Bitweise Operatoren	52
2.15 instanceof-Operator	54
3 Ablaufsteuerung: if, switch, while, do...while, for, for each.....	58
3.1 Ablaufsteuerung in Java	58
3.2 if-Anweisung	60
3.3 Alternative zur if-Anweisung	62
3.4 switch-Anweisung	64
3.5 while-Schleife	66
3.6 do...while-Schleife	66
3.7 for-Schleife	68
3.8 for each-Schleife.....	70

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

3.9 Geschachtelte Schleifen	72
3.10 Verwendung von Enhanced for Loop.....	74
3.11 break-Anweisung	76
3.12 continue-Anweisung	78
4 Klassen und Objekte	82
4.1 Objektorientierung in Java	82
4.2 Klassen	82
4.3 Attribute und Methoden	84
4.4 Syntax einer Klassendefinition.....	86
4.5 Von Klassen zu Objekten.....	88
4.6 new-Operator	88
4.7 Konstruktor	90
4.8 Definition von Konstruktoren.....	90
4.9 Zugriff auf Datentypen	92
4.10 this-Referenz	96
4.11 Instanzvariablen	98
4.12 Instanzmethoden	98
4.13 Eigenschaften von Methoden	100
4.14 Beispiel Parameterübergabe	102
4.15 return-Anweisung.....	102
4.16 Methodenaufruf mit varargs.....	104
5 Weiteres zu Klassen und Objekten	108
5.1 Vererbung	108
5.2 Beispiel: Vererbung.....	110
5.3 Überschreiben von Methoden.....	112
5.4 Beispiel: Überschreiben	112
5.5 super-Zugriff auf Superklasse.....	114
5.6 Beispiel: super	116
5.7 static-Klassenvariablen	118
5.8 static-Klassenmethoden	118
5.9 Beispiel: static.....	120
5.10 final-Attribute.....	122
5.11 Constant Interfaces	124
5.12 Zugriff auf statische Member bisher.....	124
5.13 Static Import Deklaration.....	126
5.14 Vorrang-Regelung.....	130
5.15 Nutzung von Static Import.....	132
5.16 Pakete.....	134
5.17 Einbinden von Paketen	134
5.18 Finden von Paketen	136
5.19 Erstellung eigener Pakete.....	136
5.20 Zugriffsmodifikatoren in Java.....	138
5.21 Garbage Collection	142
5.22 finalize()-Methode	142
6 Abstrakte Klassen und Interfaces	146
6.1 Abstrakte Klassen	146
6.2 Abstrakte Methoden.....	146
6.3 Abstrakte Klassen: Vererbung	148
6.4 Wozu abstrakte Klassen	150
6.5 Beispiel	150

6.6 Verwendung als Typ	152
6.7 Polymorphie.....	154
6.8 Polymorphie: Beispiel mit abstrakten Klassen.....	154
6.9 Interfaces	156
6.10 Interfaces: Member Modifikatoren	156
6.11 Weiteres zu Interfaces	158
6.12 Interfaces implementieren	158
6.13 Methoden überschreiben	160
6.14 Interfaces als Typ	162
6.15 Interfaces als Typ, Beispiel	162
6.16 Anwendungsbeispiel (1)	164
6.17 Anwendungsbeispiel (2)	164
6.18 Anwendungsbeispiel (3)	166
6.19 Anwendungsbeispiel (4)	166
7 Array, String, StringBuffer, Wrapper, Auto(un)boxing.....	170
7.1 Arrays	170
7.2 Erzeugen von Array-Objekten	172
7.3 Arrays einfacher Werte	174
7.4 Arrays von Objekten	174
7.5 Multidimensionale Arrays	176
7.6 Bemerkung zu multidimensionalen Arrays	178
7.7 Weiteres zu Arrays	180
7.8 Klasse String.....	182
7.9 Einige String-Methoden	184
7.10 Konkatenationsoperator für Strings	184
7.11 Klasse StringBuffer	186
7.12 Einige StringBuffer-Methoden.....	186
7.13 Wrapper-Klassen für primitive Typen.....	188
7.14 Beispiele mit der Klasse Integer	188
7.15 Eigenschaften von Float und Double.....	190
7.16 Boxing und Unboxing	192
7.17 Manuelles vs. automatisches Boxing	194
7.18 Autoboxing und Arrays speziell	196
7.19 Nutzung von Auto(un)boxing	198
7.20 Achtung: Referenzgleichheit.....	200
8 Nützliche Klassen	204
8.1 Klasse Vector.....	204
8.2 Beispiel mit Vector-Objekt	206
8.3 Klasse Hashtable.....	208
8.4 Beispiel mit Hashtable-Objekt.....	208
8.5 Interface Enumeration	210
8.6 Beispiel mit Enumeration-Objekt	210
8.7 Klasse Random	212
8.8 Beispiel mit Random-Objekt	212
8.9 Klasse Runtime	214
8.10 Beispiel: Externe Programme starten.....	216
8.11 Beispiel: Freien Speicher ermitteln	216
9 Ausnahmebehandlung-Exception Handling	220
9.1 Ausnahme (Exception)	220
9.2 Exception-Klassen	220
9.3 Vererbungshierarchie	222

9.4 Exception Handling.....	224
9.5 finally-Klausel.....	226
9.6 Auswerten von Ausnahmen.....	228
9.7 Auslösen von Exceptions.....	230
9.8 Definition eigener Exceptions	230
9.9 Tips	232
10 Input/Output	236
10.1 Ein- und Ausgabe in Java.....	236
10.2 Streams.....	236
10.3 Wichtige Streams.....	238
10.4 Klasse File	240
10.5 Anlegen und Schreiben einer Datei	242
10.6 Beispiel: Anlegen und Schreiben einer Datei	242
10.7 Einlesen einer Datei.....	244
10.8 Beispiel: Einlesen einer Datei	244
10.9 Löschen einer Datei.....	246
10.10 Beispiel: Löschen einer Datei	246
10.11 Objekt Serialisierung.....	248
10.12 Beispiel: Serialisierung eines Objekts.....	250
10.13 Beispiel: Deserialisierung eines Objekts.....	252
10.14 Neues I/O in Java 1.4	254
10.15 NIO, Buffer	256
10.16 NIO, Beispiel: MappedByteBuffer	256
10.17 NIO, Channel	258
11 Graphische Oberflächen mit Swing	262
11.1 Graphikbibliotheken in Java.....	262
11.2 Paket java.awt.....	264
11.3 Paket javax.swing	264
11.4 Ein einfaches Fenster mit Swing.....	266
11.5 Komponenten und Container	268
11.6 Beispiel mit Komponenten und Containern.....	268
11.7 Komponenten in Swing	270
11.8 Container in Swing.....	270
11.9 Komponenten in Container einfügen.....	272
11.10 Layout-Manager	274
11.11 BorderLayout, GridLayout	276
11.12 Fenster schließen.....	278
11.13 Event Model	280
11.14 Registrieren als Event-Listener	280
11.15 Beispiel: Fenster schließen.....	282
11.16 Events, Listener, Adapter und Methoden	284
11.17 Interaktion zwischen Komponenten	286
11.18 Beispiel mit Event-Listnern.....	286
12 Threads	290
12.1 Threads.....	290
12.2 Threads starten - 1. Möglichkeit.....	292
12.3 Threads starten - 2. Möglichkeit.....	292
12.4 Beispiel mit Threads	294
12.5 Threads kontrollieren	296
12.6 Lebensdauer eines Threads	296

12.7 Probleme mit Threads	298
12.8 Schlüsselwort synchronized	298
12.9 Beispiel mit synchronisierten Methoden	300
12.10 Synchronisierte Methoden	302
12.11 Synchronisierte Anweisungen.....	302
12.12 wait(), notify() und notifyAll()	304