

Auf einen Blick

Vorwort	15
Kapitel 1 Einleitung	19
Kapitel 2 Ein Blick auf die Entwicklungstools und erste Beispiele	39
Kapitel 3 Einführung in die objektorientierte Programmierung mit Java	81
Kapitel 4 Die grundlegenden Sprachelemente von Java ..	199
Kapitel 5 Datenstrukturen	277
Kapitel 6 Erweiterte Java-Techniken	321
Kapitel 7 Nützliches und Praktisches	355
Anhang	425
Stichwortverzeichnis	451

Inhaltsverzeichnis

Vorwort	15
Kapitel 1 Einleitung	19
1.1 Was ist Java?	20
1.1.1 Die Java Technology	20
1.1.2 Die Programmiersprache Java	20
1.1.3 Etwas zur Historie von Java	22
1.1.4 Wo findet Java heutzutage Anwendung?	25
1.1.5 Die Java-Plattform	26
1.1.6 Java und Open Source	26
1.2 Über das Buch	28
1.2.1 An wen sich das Buch wendet	29
1.2.2 Wie dieses Buch organisiert ist	29
1.2.3 Schreibkonventionen	30
1.2.4 Was Sie in dem Buch lernen	31
1.3 Was Sie unbedingt haben sollten	31
1.3.1 Die Hardware und das Betriebssystem .	31
1.3.2 Die Software	32
1.4 Wer ich bin	36
1.5 Zusammenfassung	37
Kapitel 2 Ein Blick auf die Entwicklungstools und erste Beispiele	39
2.1 Was ist ein Programm und was heißt Programmieren?	39
2.1.1 Von der Maschinensprache zur höheren Programmiersprache	40
2.1.2 Vom Quelltext zum Programm	40
2.2 Das Java Development Kit	41
2.2.1 Die Programme des JDK	41
2.2.2 Die Bibliotheken des JDK und die JRE	47
2.3 Ab ins Wasser – die ersten Beispiele	47
2.3.1 Grundlegende Regeln für Java	47
2.3.2 Das erste Beispiel – ohne IDE	49

2.3.3	Beispiel 2 – Entgegennahme von Aufrufargumenten an ein Java-Programm	55
2.3.4	Eine grafische Oberfläche als drittes Beispiel – mit Eclipse	56
2.3.5	Ein tieferer Blick in Eclipse	64
2.3.6	Eine Konsolenanwendung mit NetBeans	72
2.3.7	Eine grafische Oberfläche mit NetBeans und dessen GUI-Builder	75
2.4	Praxis und Aufgaben	79
2.4.1	Aufgaben	80
2.5	Zusammenfassung	80

Kapitel 3 Einführung in die objektorientierte Programmierung mit Java

	Kapitel 3 Einführung in die objektorientierte Programmierung mit Java	81
3.1	Objektorientierte Programmiersprachen	82
3.1.1	Bessere Software-Qualität durch OOP ..	83
3.1.2	Kernkonzepte der Objektorientierung ..	83
3.1.3	Vertragsbasierte Programmierung	84
3.2	Objekte und Klassen	85
3.2.1	Objekte	85
3.2.2	Objektorientierte Denkweise in der Benutzerführung	86
3.2.3	Objektorientierte Denkweise in der Programmierung	87
3.2.4	Klassen	90
3.2.5	Konkret in Java Klassen schreiben	93
3.2.6	Klassenelemente verwenden	106
3.2.7	Konstruktoren und Destruktoren – Objekte erzeugen und beseitigen	107
3.2.8	Fremde Klassen verwenden	117
3.3	Pakete und die import-Anweisung	118
3.3.1	Die Zuordnung einer Klasse zu einem Paket und das Default-Paket	119
3.3.2	Namenskonventionen und Standardpakete	128
3.3.3	Die Suche nach Paketen	129
3.3.4	Importieren von Klassen	131
3.3.5	Namensräume bzw. Gültigkeitsbereiche ..	135
3.4	Vererbung	136
3.4.1	Warum Vererbung? Der konkrete Nutzen	137
3.4.2	Superklasse und Subklasse	137
3.4.3	Generalisierung und Spezialisierung ..	138

3.4.4	Die technische Umsetzung einer Vererbung	139
3.4.5	Mehrfachvererbung versus Einfachvererbung	140
3.4.6	Die konkrete Umsetzung der Vererbung in Java	140
3.4.7	Überschreiben von Methoden	146
3.4.8	Zuweisungen zwischen Super- und Subklasse	151
3.4.9	Definieren von parametrisierten Konstruktoren	153
3.4.10	Überladen	154
3.5	Information Hiding und Zugriffsschutz	157
3.5.1	Die konkrete Umsetzung in Java	158
3.5.2	Lokale und anonyme Klassen	161
3.6	Indirekte Zugriffe über Getter und Setter	164
3.6.1	Die tatsächliche Notation von Getter und Setter	164
3.7	Abstrakte Klassen und Schnittstellen	167
3.7.1	Was ist eine abstrakte Klasse?	167
3.7.2	Schnittstellen	171
3.7.3	Adapterklassen	177
3.8	Weiterentwicklung des Bauernhof-API-Projekts	178
3.8.1	Schritt 1	178
3.8.2	Schritt 2 – Schnittstellen und abstrakte Klassen einsetzen	185
3.9	Praxis und Aufgaben	186
3.9.1	Praxis	186
3.9.2	Aufgaben	197
3.10	Zusammenfassung	198
Kapitel 4	Die grundlegenden Sprachelemente von Java ...	199
4.1	Token und Parser	199
4.2	Bezeichner	200
4.3	Schlüsselwörter	201
4.4	Trennzeichen und Leerzeichen	202
4.4.1	Trennzeichen	202
4.4.2	Leerzeichen	203
4.5	Literale	203
4.5.1	Ganzzahlliterale	204
4.5.2	Gleitpunktliterale	205
4.5.3	Zeichenliterale	207
4.5.4	Zeichenkettenliterale	208
4.5.5	Boolesche Literale	210
4.5.6	Binäre Literale in Java 7	210

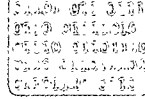
4.6	Datentypen	211
4.6.1	Die primitiven Datentypen von Java ..	211
4.6.2	Referenztypen	214
4.7	Variablen und Konstanten	214
4.7.1	Instanz- und Klassenvariablen	215
4.7.2	Lokale Variablen	215
4.7.3	Konstanten	217
4.8	Operatoren	218
4.8.1	Arithmetische Operatoren	218
4.8.2	Zuweisungsoperatoren	222
4.8.3	Vergleichsoperatoren und Referenzvergleiche	222
4.8.4	Bitweise Operatoren	230
4.8.5	Weitere Operatoren	237
4.8.6	Die Operatorpriorität	240
4.9	Typumwandlungen und Operationen mit den Datentypen	241
4.9.1	Operationen mit den Datentypen	241
4.9.2	Typkonvertierungen	246
4.10	Ausdrücke	254
4.10.1	Bewertung von Ausdrücken	255
4.11	Anweisungen	257
4.11.1	Blockanweisung	257
4.11.2	Deklarationsanweisung	257
4.11.3	Ausdrucksanweisung	258
4.11.4	Leere Anweisung	258
4.11.5	Bezeichnete Anweisung – Sprungmarken	259
4.11.6	Auswahanweisung	259
4.11.7	Iterationsanweisung	264
4.11.8	Sprunganweisung	268
4.11.9	Synchronisationsanweisungen	271
4.11.10	Schutzanweisung	271
4.11.11	Unerreichbare Anweisung	272
4.12	Praxis und Aufgaben	272
4.12.1	Aufgaben	275
4.13	Zusammenfassung	276

Kapitel 5 Datenstrukturen 277

5.1	Arrays (Datenfelder)	277
5.1.1	Einige grundsätzliche Anmerkungen zu Arrays	278
5.1.2	Deklarieren von Arrays	278
5.1.3	Erstellen von Arrays	279
5.1.4	Speichern und Zugreifen auf Array-Elemente	281

5.1.5	Spezielle Methoden zur Arbeit mit Arrays	285
5.1.6	Weiterentwicklung des Bauernhof-API-Projekts	286
5.2	Aufzählungstypen – Enums	286
5.2.1	Syntax für die Erstellung eines Enum ..	287
5.2.2	Aufzählungstypen verwenden	287
5.2.3	Attribute und Methoden bei Aufzählungstypen	289
5.2.4	Über ein Enum iterieren	290
5.3	Generics	291
5.3.1	Generische Klassen	292
5.3.2	Generische Methoden	294
5.3.3	Der Diamond-Operator	295
5.3.4	Generische Arrays	295
5.3.5	Typeinschränkungen	295
5.3.6	Wildcards	295
5.4	Allgemeine Datenstrukturen mit Container-Funktionalität – Collections & Co	298
5.4.1	Die Schnittstelle Collection	299
5.4.2	Wichtige Schnittstellen im Collection-API	302
5.4.3	Konkrete Klassen	304
5.4.4	Beispiele mit dem Collection-API	305
5.5	String, StringBuffer und StringBuilder	315
5.6	Praxis und Aufgaben	317
5.6.1	Aufgaben	319
5.7	Zusammenfassung	319
Kapitel 6	Erweiterte Java-Techniken	321
6.1	Ausnahmebehandlung	321
6.1.1	Was sind Ausnahmen?	321
6.1.2	Warum ein Ausnahmekonzept?	322
6.1.3	Die Klassen Throwable, Error und Exception	324
6.1.4	Ausnahmen behandeln	325
6.1.5	Welche Ausnahmen müssen zwingend aufgefangen werden?	326
6.1.6	Explizites Ausnahmen-Handling	327
6.1.7	Zwei praktische Beispiele	329
6.1.8	Neuerungen in Java 7	332
6.1.9	Weiterentwicklung des Bauernhof-API-Projekts	333
6.2	Assertions	335

6.3	Fabrikmethoden (Factory) und Fabrikklassen ..	338
6.3.1	Ein erstes Beispiel für eine Factory-Methode	339
6.3.2	Eine erweiterte Logik für das Beispiel	340
6.3.3	Fabrikklassen	341
6.4	Multithreading	342
6.4.1	Die Klasse Thread	343
6.4.2	Die Schnittstelle Runnable	350
6.5	Praxis und Aufgaben	350
6.5.1	Aufgaben	354
6.6	Zusammenfassung	354
Kapitel 7	Nützliches und Praktisches	355
7.1	Datum und Uhrzeit	355
7.2	Zufallszahlen erzeugen	358
7.3	Systemzugriffe	358
7.3.1	Zugriff auf Systemeigenschaften und Umgebungsvariablen	358
7.3.2	Ein Programm beenden und andere Applikationen starten	360
7.4	Grafische Benutzeroberflächen in Java	361
7.4.1	Das AWT versus Swing	362
7.4.2	Das Konzept einer grafischen Oberfläche	364
7.4.3	Ein Beispiel	369
7.5	Grafik unter Java	371
7.5.1	Allgemeines zum Zeichnen und Malen auf grafischen Java-Oberflächen	371
7.5.2	Linien und Rechtecke	372
7.5.3	Zeichnen von Kreisen und Ellipsen ...	374
7.5.4	Treiben wir es bunt – Farben unter Java	375
7.5.5	Die Textausgabe auf grafischen Oberflächen	377
7.5.6	Bilder ausgeben	379
7.5.7	Java-2D	382
7.6	Ein- und Ausgabe in Java	387
7.6.1	Die Klasse InputStream	388
7.6.2	Die Klasse OutputStream	389
7.6.3	Gefilterte Ströme	389
7.6.4	Datenströme	390
7.6.5	Die PrintStream-Klasse	391
7.6.6	Dateiarbeit und die File-Klasse	391
7.6.7	Dateiströme – FileInputStream und FileOutputStream	393



7.6.8	Die Klasse FileDescriptor	393
7.6.9	Die RandomAccessFile-Klasse	394
7.6.10	Dateizugriffe und andere Praxis mit Strömen	394
7.6.11	Allgemeine Netzwerkzugriffe	404
7.7	Die Dokumentation und Java-Code selbst dokumentieren	406
7.7.1	Die Java-API-Spezifikation	406
7.7.2	Dokumentationen erstellen	407
7.8	Verteilen und Starten von Java-Applikationen ..	409
7.8.1	Erstellen von JAR-Dateien	409
7.8.2	Eine native Startmöglichkeit bereitstellen	411
7.8.3	Java Web Start	412
7.8.4	Mit NetBeans die Java-Applikation zum Web Start fertig machen	414
7.9	Praxis und Aufgaben	416
7.9.1	Aufgaben	424
7.10	Zusammenfassung	424
Anhang	425
A.1	Versionsgeschichte von Java und Besonderheiten/Features	425
A.2	Quellen	435
A.3	Das Dokumentations-Tool javadoc	435
A.3.1	Die javadoc-Tags	436
A.4	Wichtige Java-Pakete	439
A.4.1	java.applet	440
A.4.2	java.awt	440
A.4.3	java.io	440
A.4.4	java.lang	440
A.4.5	java.math	443
A.4.6	java.net	444
A.4.7	java.security und Unterpakete	444
A.4.8	java.sql	444
A.4.9	java.text	444
A.4.10	java.util	444
A.4.11	javax.swing & Co	444
A.5	Von der Dezimalrechnung abweichende Rechenarten	444
A.5.1	Binärrechnung	445
A.5.2	Oktalrechnung	445
A.5.3	Hexadezimalrechnung	446

A.6	Die Theorie des Zweierkomplements bei ganzen Zahlen	448
A.7	Farbangaben in Java	449
A.7.1	Die direkte Angabe eines Farbnamens	449
A.7.2	Die Farbangabe in Form der RGB-Werte in Hexadezimalform	450
Stichwortverzeichnis		451