

Auf einen Blick

1	Einführung	21
2	Variablen und Datentypen	69
3	Entscheidungen	97
4	Wiederholungen	117
5	Klassen und Objekte	127
6	Objektorientierung	157
7	Unit Testing	191
8	Die Standardbibliothek	209
9	Fehler und Ausnahmen	245
10	Arrays und Collections	261
11	Lambda-Ausdrücke	291
12	Dateien, Streams und Reader	327
13	Multithreading	353
14	Servlets – Java im Web	385
15	Datenbanken und Entitäten	423
16	GUIs mit JavaFX	453
17	Android	515
18	Hinter den Kulissen	549
19	Und dann?	573

Inhalt

Materialien zum Buch	19
1 Einführung	21
1.1 Was ist Java?	22
1.1.1 Java – die Sprache	22
1.1.2 Java – die Laufzeitumgebung	23
1.1.3 Java – die Standardbibliothek	25
1.1.4 Java – die Community	25
1.1.5 Die Geschichte von Java	26
1.2 Die Arbeitsumgebung installieren	28
1.3 Erste Schritte in NetBeans	30
1.4 Das erste Programm	32
1.4.1 Der Programmcode	33
1.4.2 Die wichtigsten Elemente	36
1.4.3 Ausführen von der Kommandozeile aus	39
1.5 In Algorithmen denken, in Java schreiben	41
1.5.1 Beispiel 1: Fibonacci-Zahlen	42
1.5.2 Beispiel 2: Eine Zeichenkette umkehren	44
1.5.3 Algorithmisches Denken und Java	46
1.6 Die Java-Klassenbibliothek	47
1.7 Dokumentieren als Gewohnheit – Javadoc	50
1.7.1 Den eigenen Code dokumentieren	50
1.7.2 Package-Dokumentation	54
1.7.3 HTML-Dokumentation erzeugen	55
1.7.4 Was sollte dokumentiert sein?	56
1.8 JARs erstellen und ausführen	56
1.8.1 Die Datei »MANIFEST.MF«	57
1.8.2 JARs ausführen	58
1.8.3 JARs erzeugen	59
1.8.4 JARs einsehen und entpacken	60
1.9 Mit dem Debugger arbeiten	61
1.9.1 Ein Programm im Debug-Modus starten	61
1.9.2 Breakpoints und schrittweise Ausführung	61

1.9.3	Variablenwerte und Call Stack inspizieren	62
1.9.4	Übung: Der Debugger	64
1.10	Das erste eigene Projekt	66
1.11	Zusammenfassung	68

2 Variablen und Datentypen 69

2.1	Variablen	69
2.1.1	Der Zuweisungsoperator	71
2.1.2	Scopes	71
2.1.3	Das Schlüsselwort »var«	72
2.1.4	Primitive und Objekte	73
2.2	Primitivtypen	73
2.2.1	Zahlentypen	73
2.2.2	Rechenoperationen	77
2.2.3	Bit-Operatoren	81
2.2.4	Übung: Ausdrücke und Datentypen	83
2.2.5	Character-Variablen	83
2.2.6	Boolesche Variablen	85
2.2.7	Vergleichsoperatoren	85
2.3	Objekttypen	87
2.3.1	Werte und Referenzen	87
2.3.2	Der Wert »null«	88
2.3.3	Vergleichsoperatoren	89
2.3.4	Allgemeine und spezielle Typen	89
2.3.5	Strings – primitive Objekte	91
2.4	Objekt-Wrapper zu Primitiven	91
2.4.1	Warum braucht man Primitiv-Wrapper?	92
2.4.2	Explizite Konvertierung	92
2.4.3	Implizite Konvertierung	93
2.5	Array-Typen	94
2.5.1	Deklaration eines Arrays	95
2.5.2	Zugriff auf ein Array	96
2.6	Zusammenfassung	96

3 Entscheidungen

97

3.1 Entweder-oder-Entscheidungen	97
3.1.1 Übung: Star Trek – sehen oder nicht?	99
3.1.2 Mehrfache Verzweigungen	101
3.1.3 Übung: Body-Mass-Index	102
3.1.4 Der ternäre Operator	102
3.2 Logische Verknüpfungen	104
3.2.1 Boolesche Operatoren	104
3.2.2 Verknüpfungen mit und ohne Kurzschluss	105
3.2.3 Übung: Boolesche Operatoren	107
3.2.4 Übung: Solitaire	108
3.3 Mehrfach verzweigen mit »switch«	110
3.3.1 »switch« mit Strings, Zeichen und Zahlen	111
3.3.2 Übung: »Rock im ROM«	112
3.3.3 Enumerierte Datentypen und »switch«	113
3.3.4 Durchfallendes »switch«	114
3.3.5 Übung: »Rock im ROM« bis zum Ende	115
3.3.6 Übung: »Rock im ROM« solange ich will	115
3.3.7 Der Unterschied zwischen »switch« und »if... else if ...«	115
3.4 Zusammenfassung	116

4 Wiederholungen

117

4.1 Bedingte Wiederholungen mit »while«	117
4.1.1 Kopfgesteuerte »while«-Schleife	118
4.1.2 Übung: Das kleinste gemeinsame Vielfache	119
4.1.3 Fußgesteuerte »while«-Schleifen	119
4.1.4 Übung: Zahlen raten	120
4.2 Abgezählte Wiederholungen – die »for«-Schleife	121
4.2.1 Übung: Zahlen validieren	122
4.3 Abbrechen und überspringen	123
4.3.1 »break« und »continue« mit Labels	124
4.4 Zusammenfassung	126

5 Klassen und Objekte 127

5.1 Klassen und Objekte	128
5.1.1 Klassen anlegen	128
5.1.2 Objekte erzeugen	129
5.2 Access-Modifizier	130
5.3 Felder	132
5.3.1 Felder deklarieren	132
5.3.2 Zugriff auf Felder	132
5.4 Methoden	133
5.4.1 Übung: Eine erste Methode	135
5.4.2 Rückgabewerte	135
5.4.3 Übung: Jetzt mit Rückgabewerten	137
5.4.4 Parameter	137
5.4.5 Zugriffsmethoden	139
5.4.6 Übung: Zugriffsmethoden	141
5.5 Warum Objektorientierung?	142
5.6 Konstruktoren	144
5.6.1 Konstruktoren deklarieren und aufrufen	144
5.6.2 Übung: Konstruktoren	148
5.7 Statische Felder und Methoden	148
5.7.1 Übung: Statische Felder und Methoden	150
5.7.2 Die »main«-Methode	150
5.7.3 Statische Importe	150
5.8 Unveränderliche Werte	151
5.8.1 Unveränderliche Felder	152
5.8.2 Konstanten	153
5.9 Spezielle Objektmethoden	154
5.10 Zusammenfassung	156

6 Objektorientierung 157

6.1 Vererbung	158
6.1.1 Vererbung implementieren	159
6.1.2 Übung: Tierische Erbschaften	161

6.1.3	Erben und Überschreiben von Mitgliedern	161
6.1.4	Vererbung und Konstruktoren	166
6.1.5	Übung: Konstruktoren und Vererbung	167
6.1.6	Vererbung verhindern	167
6.1.7	Welchen Typ hat das Objekt?	169
6.2	Interfaces und abstrakte Datentypen	171
6.2.1	Abstrakte Klassen	172
6.2.2	Interfaces	173
6.2.3	Default-Implementierungen	176
6.3	Übung: Objektorientierte Modellierung	179
6.4	Innere Klassen	180
6.4.1	Statische innere Klassen	180
6.4.2	Nichtstatische innere Klassen	182
6.4.3	Anonyme Klassen	185
6.5	Enumerationen	187
6.6	Zusammenfassung	190
7	Unit Testing	191
<hr/>		
7.1	Das JUnit-Framework	193
7.1.1	Der erste Test	194
7.1.2	Die Methoden von »Assert«	196
7.1.3	Testfälle ausführen in NetBeans	196
7.1.4	Übung: Den GGT-Algorithmus ändern	199
7.1.5	Übung: Tests schreiben für das KGV	199
7.2	Fortgeschrittene Unit Tests	199
7.2.1	Testen von Fehlern	200
7.2.2	Vor- und Nachbereitung von Tests	201
7.2.3	Mocking	203
7.3	Besseres Design durch Testfälle	205
7.3.1	Übung: Testfälle für den BMI-Rechner	208
7.4	Zusammenfassung	208

8 Die Standardbibliothek

209

8.1 Zahlen	209
8.1.1 »Number« und die Zahlentypen	209
8.1.2 Mathematisches aus »java.lang.Math«	210
8.1.3 Übung: Satz des Pythagoras	213
8.1.4 »BigInteger« und »BigDecimal«	213
8.1.5 Übung: Fakultäten	214
8.2 Strings	214
8.2.1 Unicode	215
8.2.2 String-Methoden	216
8.2.3 Übung: Namen zerlegen	220
8.2.4 Übung: Römische Zahlen I	220
8.2.5 StringBuilder	221
8.2.6 Übung: Römische Zahlen II	223
8.2.7 StringTokenizer	224
8.3 Reguläre Ausdrücke	224
8.3.1 Einführung in reguläre Ausdrücke	224
8.3.2 String-Methoden mit regulären Ausdrücken	227
8.3.3 Reguläre Ausdrücke als Objekte	228
8.3.4 Übung: Flugnummern finden	231
8.4 Zeit und Datum	231
8.4.1 Zeiten im Computer und »java.util.Date«	231
8.4.2 Neue Zeiten – das Package »java.time«	232
8.4.3 Übung: Der Fernsehkalender	236
8.5 Internationalisierung und Lokalisierung	236
8.5.1 Internationale Nachrichten mit »java.util.ResourceBundle«	237
8.5.2 Nachrichten formatieren mit »java.util.MessageFormat«	239
8.5.3 Zeiten und Daten lesen	242
8.5.4 Zahlen lesen	243
8.6 Zusammenfassung	244

9 Fehler und Ausnahmen

245

9.1 Exceptions werfen und behandeln	245
9.1.1 try-catch	247
9.1.2 Übung: Fangen und noch einmal versuchen	249

9.1.3	try-catch-finally	249
9.1.4	try-with-resources	251
9.1.5	Fehler mit Ursachen	251
9.2	Verschiedene Arten von Exceptions	252
9.2.1	Unchecked Exceptions	253
9.2.2	Checked Exceptions	254
9.2.3	Errors	257
9.3	Invarianten, Vor- und Nachbedingungen	258
9.4	Zusammenfassung	260

10 Arrays und Collections 261

10.1	Arrays	261
10.1.1	Grundlagen von Arrays	262
10.1.2	Übung: Primzahlen	264
10.1.3	Mehrdimensionale Arrays	265
10.1.4	Übung: Das Pascalsche Dreieck	266
10.1.5	Utility-Methoden in »java.util.Arrays«	266
10.1.6	Übung: Sequenziell und parallel sortieren	270
10.2	Die for-each-Schleife	271
10.3	Variable Parameterlisten	271
10.4	Collections	273
10.4.1	Listen und Sets	274
10.4.2	Iteratoren	277
10.4.3	Übung: Musiksammlung und Playlist	278
10.5	Typisierte Collections – Generics	278
10.5.1	Generics außerhalb von Collections	280
10.5.2	Eigenen Code generifizieren	282
10.5.3	Übung: Generisches Filtern	288
10.6	Maps	288
10.6.1	Übung: Lieblingslieder	290
10.7	Zusammenfassung	290

11 Lambda-Ausdrücke 291

11.1 Was sind Lambda-Ausdrücke?	292
11.1.1 Die Lambda-Syntax	293
11.1.2 Wie funktioniert das?	296
11.1.3 Übung: Zahlen selektieren	299
11.1.4 Funktionale Interfaces nur für Lambda-Ausdrücke	299
11.1.5 Übung: Funktionen	304
11.2 Die Stream-API	304
11.2.1 Intermediäre und terminale Methoden	306
11.2.2 Übung: Temperaturdaten auswerten	316
11.2.3 Endlose Streams	316
11.2.4 Übung: Endlose Fibonacci-Zahlen	317
11.2.5 Daten aus einem Stream sammeln – »Stream.collect«	318
11.2.6 Übung: Wetterstatistik für Fortgeschrittene	321
11.3 Un-Werte als Objekte – »Optional«	321
11.3.1 Die wahre Bedeutung von »Optional«	323
11.4 Eine Warnung zum Schluss	324
11.5 Zusammenfassung	325

12 Dateien, Streams und Reader 327

12.1 Dateien und Verzeichnisse	328
12.1.1 Dateien und Pfade	328
12.1.2 Dateioperationen aus »Files«	331
12.1.3 Übung: Dateien kopieren	331
12.1.4 Verzeichnisse	332
12.1.5 Übung: Musik finden	333
12.2 Reader, Writer und die »anderen« Streams	334
12.2.1 Lesen und Schreiben von Textdaten	335
12.2.2 Übung: Playlisten – jetzt richtig	342
12.2.3 »InputStream« und »OutputStream« – Binärdaten	342
12.2.4 Übung: ID3-Tags	344
12.3 Objekte lesen und schreiben	346
12.3.1 Serialisierung	346

12.4	Netzwerkkommunikation	349
12.4.1	Übung: Dateitransfer	351
12.5	Zusammenfassung	352

13 Multithreading 353

13.1	Threads und Runnables	354
13.1.1	Threads starten und Verhalten übergeben	354
13.1.2	Übung: Multithreaded Server	358
13.1.3	Geteilte Ressourcen	358
13.2	Atomare Datentypen	361
13.3	Synchronisation	362
13.3.1	»synchronized« als Modifikator für Methoden	364
13.3.2	Das »synchronized«-Statement	365
13.3.3	Deadlocks	368
13.3.4	Übung: Zufallsverteilung	369
13.4	Fortgeschrittene Koordination zwischen Threads	370
13.4.1	Signalisierung auf dem Monitor-Objekt	370
13.4.2	Daten produzieren, kommunizieren und konsumieren	373
13.4.3	Threads wiederverwenden	375
13.5	Die Zukunft – wortwörtlich	376
13.5.1	Lambdas und die Zukunft – »CompletableFuture«	379
13.6	Das Speichermodell von Threads	381
13.7	Zusammenfassung	383

14 Servlets – Java im Web 385

14.1	Einen Servlet-Container installieren	386
14.1.1	Installation des Tomcat-Servers	386
14.1.2	Den Tomcat-Server in NetBeans einrichten	390
14.2	Die erste Servlet-Anwendung	392
14.2.1	Die Anwendung starten	394
14.2.2	Was geschieht, wenn Sie die Anwendung aufrufen?	397
14.3	Servlets programmieren	403
14.3.1	Servlets konfigurieren	403

14.3.2	Mit dem Benutzer interagieren	405
14.3.3	Übung: Das Rechen-Servlet implementieren	408
14.4	Java Server Pages	410
14.4.1	Übung: Playlisten anzeigen	415
14.4.2	Übung: Musik abspielen	415
14.5	Langlebige Daten im Servlet – Ablage in Session und Application	416
14.5.1	Die »HTTPSession«	417
14.5.2	Übung: Daten in der Session speichern	418
14.5.3	Der Application Context	418
14.6	Fortgeschrittene Servlet-Konzepte – Listener und Initialisierung	419
14.6.1	Listener	419
14.6.2	Übung: Die Playliste nur einmal laden	420
14.6.3	Initialisierungsparameter	420
14.7	Zusammenfassung	422
15	Datenbanken und Entitäten	423
<hr/>		
15.1	Was ist eine Datenbank?	424
15.1.1	Relationale Datenbanken	424
15.1.2	JDBC	428
15.1.3	JPA	429
15.2	Mit einer Datenbank über die JPA verbinden	431
15.2.1	Datenbank in NetBeans anlegen	431
15.2.2	Das Projekt anlegen	432
15.2.3	Eine Persistence Unit erzeugen	433
15.2.4	Die »EntityManagerFactory« erzeugen	435
15.3	Anwendung und Entitäten	436
15.3.1	Die erste Entität anlegen	436
15.3.2	Übung: Personen speichern	439
15.4	Entitäten laden	439
15.4.1	Abfragen mit JPQL	439
15.4.2	Übung: Personen auflisten	441
15.4.3	Entitäten laden mit ID	442
15.4.4	Übung: Personen bearbeiten	442
15.4.5	Benannte Queries	443
15.5	Entitäten löschen	444

15.6 Beziehungen zu anderen Entitäten	445
15.6.1 Eins-zu-eins-Beziehungen	446
15.6.2 Übung: Kontakte mit Adressen	448
15.6.3 Eins-zu-viele-Beziehungen	448
15.6.4 Viele-zu-eins-Beziehungen	449
15.6.5 Beziehungen in JPQL	451
15.7 Zusammenfassung	452

16 GUIs mit JavaFX 453

16.1 Einführung	453
16.2 Installation	454
16.3 Architektur von JavaFX	454
16.3.1 Application	455
16.3.2 Scenes	456
16.3.3 Scene Graph	456
16.3.4 Typen von Nodes	457
16.4 GUI-Komponenten	457
16.4.1 Beschriftungen	458
16.4.2 Schaltflächen	458
16.4.3 Checkboxes und Choiceboxes	460
16.4.4 Eingabefelder	462
16.4.5 Menüs	462
16.4.6 Sonstige Standardkomponenten	464
16.4.7 Geometrische Komponenten	467
16.4.8 Diagramme	467
16.5 Layouts	468
16.5.1 BorderPane	468
16.5.2 HBox	470
16.5.3 VBox	470
16.5.4 StackPane	471
16.5.5 GridPane	472
16.5.6 FlowPane	473
16.5.7 TilePane	474
16.5.8 AnchorPane	475
16.5.9 Fazit	477
16.6 GUI mit Java-API – Urlaubsverwaltung	478
16.6.1 Initialisierung des Menüs	479

16.6.2	Initialisierung der Tabs	479
16.6.3	Initialisierung des Inhalts von Tab 1	479
16.6.4	Initialisierung des Inhalts von Tab 2	481
16.7	Event-Handling	482
16.7.1	Events und Event-Handler	483
16.7.2	Typen von Events	485
16.7.3	Alternative Methoden für das Registrieren von Event-Handlern	488
16.8	JavaFX-Properties und Binding	489
16.8.1	JavaFX-Properties	490
16.8.2	JavaFX-Properties und Listener	491
16.8.3	JavaFX-Properties im GUI	492
16.8.4	JavaFX-Properties von GUI-Komponenten	493
16.8.5	Binding	494
16.9	Deklarative GUIs mit FXML	495
16.9.1	Vorteile gegenüber programmatisch erstellten GUIs	495
16.9.2	Einführung	497
16.9.3	Aufruf eines FXML-basierten GUI	499
16.9.4	Event-Handling in FXML	499
16.10	Layout mit CSS	501
16.10.1	Einführung in CSS	501
16.10.2	JavaFX-CSS	502
16.10.3	JavaFX-Anwendung mit CSS	503
16.10.4	Urlaubsverwaltung mit JavaFX-CSS	503
16.11	Transformationen, Animationen und Effekte	505
16.11.1	Transformationen	505
16.11.2	Animationen	508
16.12	Übungen	512
16.12.1	Eine kleine To-do-Anwendung	512
16.12.2	Logik für die To-do-Anwendung	513
16.13	Zusammenfassung	513

17 Android 515

17.1	Einstieg in die Android-Entwicklung	515
17.1.1	Die Entwicklungsumgebung	516
17.1.2	Die erste Anwendung	517
17.1.3	Der Android Emulator	520

17.1.4	Auf dem Telefon ausführen	523
17.1.5	Die erste Android-Anwendung im Detail	525
17.2	Eine Benutzeroberfläche designen	528
17.2.1	Layouts bearbeiten	530
17.2.2	Auf Widgets reagieren	533
17.2.3	Das Android-Thread-Modell	534
17.2.4	Übung: Ein ganz einfacher Rechner	535
17.3	Anwendungen mit mehreren Activities	535
17.3.1	Activity wechseln mit Intents	536
17.3.2	Der Activity Stack	538
17.3.3	An andere Anwendungen verweisen	540
17.4	Permissions und SystemServices	542
17.4.1	Den Benutzer um Erlaubnis fragen	542
17.4.2	Zugriff auf einen SystemService erlangen	544
17.4.3	Den Vibrationsservice verwenden	544
17.4.4	Übung: Die Samuel-Morse-Gedenkübung	545
17.5	Apps im Play Store veröffentlichen	546
17.6	Zusammenfassung	546

18 Hinter den Kulissen 549

18.1	Klassenpfade und Classloading	549
18.1.1	Klassen laden in der Standardumgebung	550
18.1.2	Ein komplexeres Szenario – Klassen laden im Servlet-Container	551
18.1.3	ClassLoader und Klassengleichheit	552
18.1.4	ClassLoader als Objekte	554
18.1.5	Klassen laden mit Struktur: das Modulsystem von Java 9	554
18.2	Garbage Collection	556
18.2.1	Speicherlecks in Java	560
18.2.2	Weiche und schwache Referenzen	561
18.3	Flexibel codieren mit der Reflection-API	563
18.3.1	Übung: Templating	568
18.4	Blick in die Zukunft	569
18.4.1	Der switch-Ausdruck	570
18.4.2	Mehrzeilige Strings	571
18.5	Zusammenfassung	572

19 Und dann?	573
19.1 Jakarta EE	574
19.1.1 Servlet	574
19.1.2 JPA	576
19.1.3 Enterprise Java Beans	576
19.1.4 Java Messaging Service	577
19.1.5 Java Bean Validation	578
19.2 Open-Source-Software	579
19.3 Ergänzende Technologien	580
19.3.1 SQL und DDL	580
19.3.2 HTML, CSS und JavaScript	581
19.4 Andere Sprachen	583
19.4.1 Scala	583
19.4.2 Clojure	584
19.4.3 JavaScript	584
19.5 Programmieren Sie!	585
Anhang	587
A Java-Bibliotheken	589
B Lösungen zu den Übungsaufgaben	597
C Glossar	707
D Kommandozeilenparameter	723
Index	731