

Inhaltsverzeichnis

Vorwort	xv
----------------------	-----------

I Einstieg	1
1 Einführung	3
1.1 Java im Überblick	3
1.2 Los geht's – Installation	5
1.2.1 Java-Download	5
1.2.2 Installation des JDKs	7
1.2.3 Nacharbeiten nach der Java-Installation	7
1.2.4 Java-Installation prüfen	8
1.3 Entwicklungsumgebungen	9
1.3.1 Installation von Eclipse	10
1.3.2 Eclipse starten	12
1.3.3 Erstes Projekt in Eclipse	14
1.3.4 Erste Klasse in Eclipse	16
2 Schnelleinstieg	19
2.1 Hallo Welt (Hello World)	19
2.2 Variablen und Datentypen	20
2.2.1 Definition von Variablen	20
2.2.2 Bezeichner (Variablennamen)	23
2.3 Operatoren im Überblick	25
2.3.1 Arithmetische Operatoren	25
2.3.2 Zuweisungsoperatoren	28
2.3.3 Vergleichsoperatoren	30
2.3.4 Logische Operatoren	31
2.4 Fallunterscheidungen	32

2.5	Methoden	36
2.5.1	Methoden aus dem JDK nutzen	36
2.5.2	Eigene Methoden definieren	37
2.5.3	Nützliche Beispiele aus dem JDK	40
2.5.4	Signatur einer Methode	43
2.5.5	Fehlerbehandlung und Exceptions	44
2.6	Kommentare	44
2.7	Schleifen	45
2.7.1	Die <code>for</code> -Schleife	45
2.7.2	Die <code>for-each</code> -Schleife	46
2.7.3	Die <code>while</code> -Schleife	47
2.7.4	Die <code>do-while</code> -Schleife	48
2.8	Rekapitulation	49
2.9	Weiterführende Dokumentation für nächste Schritte	50
2.10	Aufgaben und Lösungen	51
2.10.1	Aufgabe 1: Mathematische Berechnungen	51
2.10.2	Aufgabe 2: Methode und <code>if</code>	51
2.10.3	Aufgabe 3: Selbstabholerrabatt	52
2.10.4	Aufgabe 4: Schleifen mit Berechnungen	53
2.10.5	Aufgabe 5: Schleifen und fixe Schrittweite	54
2.10.6	Aufgabe 6: Schleifen mit variabler Schrittweite	54
2.10.7	Aufgabe 7: Verschachtelte Schleifen – Variante 1	55
2.10.8	Aufgabe 8: Verschachtelte Schleifen – Variante 2	56
2.10.9	Aufgabe 9: Verschachtelte Schleifen – Variante 3	57
3	Strings	59
3.1	Schnelleinstieg	59
3.1.1	Gebräuchliche Stringaktionen	59
3.1.2	Suchen und Ersetzen	63
3.1.3	Informationen extrahieren und formatieren	65
3.2	Nächste Schritte	67
3.2.1	Die Klasse <code>Scanner</code>	68
3.2.2	Mehrzeilige Strings (Text Blocks)	71
3.2.3	Strings und <code>char[]</code> s	72
3.3	Praxisbeispiel: Text in Title Case wandeln	74
3.4	Aufgaben und Lösungen	76
3.4.1	Aufgabe 1: Länge, Zeichen und Enthaltensein	76
3.4.2	Aufgabe 2: Title Case mit <code>Scanner</code>	76
3.4.3	Aufgabe 3: Zeichen wiederholen	77
3.4.4	Aufgabe 4: Vokale raten	78
3.4.5	Aufgabe 5: String Merge	80

4	Arrays	83
4.1	Schnelleinstieg	83
4.1.1	Gebräuchliche Aktionen	83
4.1.2	Mehrdimensionale Arrays	89
4.2	Nächste Schritte	90
4.2.1	Eindimensionale Arrays	91
4.2.2	Mehrdimensionale Arrays	95
4.3	Praxisbeispiel: Flächen füllen	99
4.4	Aufgaben und Lösungen	102
4.4.1	Aufgabe 1: Durcheinanderwürfeln eines Arrays	102
4.4.2	Aufgabe 2: Arrays kombinieren	103
4.4.3	Aufgabe 3: Rotation um eine oder mehrere Positionen	104
4.4.4	Aufgabe 4: Zweidimensionales String-Array ausgeben	106
4.4.5	Aufgabe 5: Dreieckiges Array: Upside Down	107
5	Klassen und Objektorientierung	109
5.1	Schnelleinstieg	109
5.1.1	Grundlagen zu Klassen und Objekten	110
5.1.2	Eigenschaften (Attribute)	112
5.1.3	Verhalten (Methoden)	117
5.1.4	Objekte vergleichen – die Rolle von <code>equals()</code>	120
5.2	Nächste Schritte	123
5.2.1	Klassen ausführbar machen	123
5.2.2	Imports und Packages	126
5.2.3	Übergang zum Einsatz einer IDE	128
5.2.4	Imports und Packages: Auswirkungen auf unsere Applikation	132
5.2.5	Verstecken von Informationen	138
5.3	Vererbung	142
5.3.1	Basisklassen und abstrakte Basisklassen	143
5.3.2	Overloading und Overriding	145
5.4	Die Klasse <code>Object</code>	147
5.4.1	Beispielklasse <code>Person</code>	148
5.4.2	Die Methode <code>toString()</code>	149
5.4.3	Ergänzungen zur Methode <code>equals(Object)</code>	150
5.4.4	Typprüfung mit <code>instanceof</code>	150
5.4.5	Pattern Matching bei <code>instanceof</code>	151
5.5	Schnittstelle (Interface) und Implementierung	152
5.6	Records	154
5.7	Aufgaben und Lösungen	156
5.7.1	Aufgabe 1: Obstkorb	156
5.7.2	Aufgabe 2: Superheld	157
5.7.3	Aufgabe 3: Zähler	159
5.7.4	Aufgabe 4: Zähler mit Überlauf	161

6	Collections	165
6.1	Schnelleinstieg.....	165
6.1.1	Die Klasse <code>ArrayList</code>	165
6.1.2	Die Klasse <code>HashSet</code>	171
6.1.3	Iteratoren	175
6.1.4	Die Klasse <code>HashMap</code>	177
6.2	Nächste Schritte	181
6.2.1	Generische Typen (Generics)	181
6.2.2	Basisinterfaces für die Containerklassen	186
6.3	Praxisbeispiel: Einen Stack selbst realisieren	190
6.4	Aufgaben und Lösungen	192
6.4.1	Aufgabe 1: Tennisverein-Mitgliederliste	192
6.4.2	Aufgabe 2: Liste mit Farbnamen füllen und filtern	193
6.4.3	Aufgabe 3: Duplikate entfernen – Variante 1	193
6.4.4	Aufgabe 4: Duplikate entfernen – Variante 2	194
6.4.5	Aufgabe 5: Hauptstädte	195
6.4.6	Aufgabe 6: Häufigkeiten von Namen	195
6.4.7	Aufgabe 7: Objekte mit Maps selbst gebaut	196
6.4.8	Aufgabe 8: Listenreihenfolge umdrehen (mit Stack)	197
7	Ergänzendes Wissen	199
7.1	Sichtbarkeits- und Gültigkeitsbereiche	199
7.2	Primitive Typen und Wrapper-Klassen.....	201
7.2.1	Grundlagen	201
7.2.2	Casting: Typerverweiterungen sowie -verkleinerungen	206
7.2.3	Konvertierung von Werten.....	207
7.3	Ternary-Operator (?-Operator)	210
7.4	Aufzählungen mit <code>enum</code>	211
7.5	Switch	213
7.6	Moderne Switch Expressions	215
7.6.1	Einführendes Beispiel	215
7.6.2	Weitere Gründe für die Neuerung	217
7.7	Pattern Matching bei Switch Expressions (Java 17 Preview)	219
7.7.1	Einführendes Beispiel	219
7.7.2	Spezialitäten	220
7.8	<code>Break</code> und <code>Continue</code> in Schleifen	222
7.8.1	Funktionsweise von <code>break</code> und <code>continue</code> in Schleifen	222
7.8.2	Wie macht man es besser?	224
7.9	Rekursion	226

7.10 Aufgaben und Lösungen	229
7.10.1 Aufgabe 1: Würfelspiel	229
7.10.2 Aufgabe 2: Prüfung auf Vokale mit <code>switch</code>	230
7.10.3 Aufgabe 3: Temperaturumrechnung	230
7.10.4 Aufgabe 4: Palindrom-Prüfung mit Rekursion	232

II Aufstieg 235

8 Mehr zu Klassen und Objektorientierung	237
8.1 Wissenswertes zu Vererbung	237
8.1.1 Generalisierung und Spezialisierung	237
8.1.2 Polymorphie	239
8.1.3 Sub-Classing und Sub-Typing	240
8.2 Varianten innerer Klassen	241
8.2.1 »Normale« innere Klassen	241
8.2.2 Statische innere Klassen	242
8.2.3 Methodenlokale innere Klassen	243
8.2.4 Anonyme innere Klassen	243
9 Lambdas und Streams	245
9.1 Einstieg in Lambdas	245
9.1.1 Syntax von Lambdas	245
9.1.2 Functional Interfaces und SAM-Typen	246
9.2 Methodenreferenzen	249
9.3 Externe vs. interne Iteration	250
9.3.1 Externe Iteration	251
9.3.2 Interne Iteration	251
9.3.3 Das Interface <code>Predicate<T></code>	252
9.4 Streams im Überblick	253
9.4.1 Streams erzeugen – Create Operations	254
9.4.2 Intermediate und Terminal Operations im Überblick	255
9.4.3 Zustandslose Intermediate Operations	255
9.4.4 Zustandsbehaftete Intermediate Operations	260
9.4.5 Terminal Operations	261
9.5 Aufgaben und Lösungen	264
9.5.1 Aufgabe 1: Erwachsene aus Personenliste extrahieren	264
9.5.2 Aufgabe 2: Stream-API	265
9.5.3 Aufgabe 3: Informationen mit Stream-API extrahieren	266
9.5.4 Aufgabe 4: Häufigkeiten von Namen	267
9.5.5 Aufgabe 5: Kollektoren	267

10	Verarbeitung von Dateien	269
10.1	Schnelleinstieg	269
10.1.1	Das Interface <code>Path</code> und die Utility-Klasse <code>Files</code>	270
10.1.2	Anlegen von Dateien und Verzeichnissen	270
10.1.3	Inhalt eines Verzeichnisses auflisten	271
10.1.4	Pfad ist Datei oder Verzeichnis?	272
10.1.5	Dateiaktionen und die Utility-Klasse <code>Files</code>	272
10.1.6	Informationen zu <code>Path</code> -Objekten ermitteln	276
10.1.7	Kopieren	277
10.1.8	Umbenennen	279
10.1.9	Löschen	280
10.2	Dateibehandlung und die Klasse <code>File</code>	281
10.2.1	Konvertierung von <code>Path</code> in <code>File</code> und zurück	282
10.2.2	Die Klasse <code>File</code> im Kurzüberblick	282
10.2.3	Dateiinhalte verarbeiten und Systemressourcen	283
10.3	Praxisbeispiel: Directory-Baum darstellen	285
10.3.1	Basisvariante	286
10.3.2	Variante mit schönerer Darstellung	288
10.3.3	Finale Variante mit ausgeklügelter Darstellung	289
10.4	Aufgaben und Lösungen	291
10.4.1	Aufgabe 1: Texte in Datei schreiben und wieder lesen	291
10.4.2	Aufgabe 2: Dateigrößen	291
10.4.3	Aufgabe 3: Existenzprüfung	292
10.4.4	Aufgabe 4: Rechteprüfung	292
10.4.5	Aufgabe 5: Verzeichnisinhalt auflisten	293
11	Fehlerbehandlung mit Exceptions	295
11.1	Schnelleinstieg	295
11.1.1	Fehlerbehandlung	296
11.1.2	Exceptions selbst auslösen – <code>throw</code>	301
11.1.3	Eigene Exception-Typen definieren	302
11.1.4	Exceptions propagieren – <code>throws</code>	302
11.2	Fehlerbehandlung in der Praxis	303
11.3	Automatic Resource Management (ARM)	305
11.4	Hintergrundwissen: Checked und Unchecked Exceptions	306
12	Datumsverarbeitung	309
12.1	Schnelleinstieg	309
12.1.1	Die Aufzählungen <code>DayOfWeek</code> und <code>Month</code>	309
12.1.2	Die Klasse <code>LocalDate</code>	311
12.1.3	Die Klassen <code>LocalTime</code> und <code>LocalDateTime</code>	315

12.2 Nächste Schritte	317
12.2.1 Datumsarithmetik	317
12.2.2 Formatierung und Parsing	319
12.3 Praxisbeispiel: Kalenderausgabe	321
12.4 Aufgaben und Lösungen	324
12.4.1 Aufgabe 1: Wochentage	324
12.4.2 Aufgabe 2: Freitag, der 13.	325
12.4.3 Aufgabe 3: Mehrmals Freitag, der 13.	326
12.4.4 Aufgabe 4: Schaltjahre	327

III **Praxisbeispiele** **329**

13 Praxisbeispiel: Tic Tac Toe	331
13.1 Spielfeld initialisieren und darstellen	331
13.2 Setzen der Steine	332
13.3 Prüfen auf Sieg	333
13.4 Bausteine im Einsatz	335
14 Praxisbeispiel: CSV-Highscore-Liste einlesen	337
14.1 Verarbeitung von Spielständen (Highscores)	337
14.2 Extraktion der Daten	339
14.3 Besonderheiten der Implementierung	340
15 Praxisbeispiel: Worträtsel	341
15.1 Applikationsdesign – Vorüberlegungen zur Strukturierung	342
15.2 Einlesen der verfügbaren Wörter	342
15.3 Hilfsdatenstrukturen	344
15.4 Datenmodell	344
15.4.1 Datenspeicherung und Initialisierung	344
15.4.2 Zufällige Wahl von Richtung, Position, Wort und Buchstabe ..	346
15.4.3 Algorithmus zum Verstecken von Wörtern	346
15.4.4 Wort prüfen und platzieren	348
15.5 HTML-Erzeugung	350
15.6 Hauptapplikation	351
15.7 Ausgabe als HTML und Darstellung im Browser	353
15.8 Fazit	354

IV Schlussgedanken **355**

16 Gute Angewohnheiten	357
16.1 Grundregeln eines guten Programmierstils	357
16.2 Coding Conventions	358
16.2.1 Grundlegende Namens- und Formatierungsregeln	358
16.2.2 Namensgebung	360
16.2.3 Dokumentation	363
16.2.4 Programmdesign	363
16.2.5 Parameterlisten	364
16.2.6 Logik und Kontrollfluss	365
16.3 Sourcecode-Prüfung	366
16.4 JUnit 5: Auch ans Testen denken	368
16.4.1 Das JUnit-Framework	368
16.4.2 Schreiben und Ausführen von Tests	370
17 Schlusswort	373

V Anhang **375**

A Schlüsselwörter im Überblick	377
B Schnelleinstieg JShell	381
C Grundlagen zur JVM und Infos zum Java-Ökosystem	385
C.1 Wissenswertes zur Java Virtual Machine (JVM)	385
C.1.1 Einführendes Beispiel	386
C.1.2 Ausführung eines Java-Programms	387
C.2 Das Java-Ökosystem im Kurzüberblick	388
Literaturverzeichnis	391
Index	393