

Inhalt

Vorwort	21
1 Hello World!	23
<hr/>	
1.1 Einführung	23
Warum Java?	24
Das große Versionsnummernchaos	25
Kleines Java-Glossar	27
Windows, Linux oder macOS?	28
1.2 Java und die IntelliJ IDEA installieren	29
1.3 Installation unter Windows	30
Path-Variable einstellen	30
Notepad++ und IntelliJ IDEA installieren	32
1.4 Installation unter Ubuntu Linux	33
IntelliJ IDEA installieren	33
1.5 Installation unter macOS	34
IntelliJ IDEA installieren	34
1.6 »Hello World« mit javac und java manuell übersetzen	34
Code verfassen und speichern	34
Das Programm kompilieren und ausführen	36
Der Hello-World-Code	38
Zulässige Codeänderungen	41
Java-Interna	42
1.7 Die Java-Shell	44
Fortgeschrittene Funktionen	45

1.8	Hello IntelliJ IDEA	46
	JDK einrichten	47
	Neues Projekt starten	48
	Endlich Code	49
	Beispieldateien	50
	IntelliJ(gent) oder nicht – das ist hier die Frage	51
1.9	Wiederholungsfragen	52
2	Java-Crashkurs	53
<hr/>		
2.1	Die Idee des objektorientierten Programmierens	53
	Methoden helfen, Teilaufgaben zu lösen	54
	Klassen bringen Daten und Methoden zusammen	54
	Objekte sind konkrete Ausformungen von Klassen	55
	Begriffe	56
2.2	Java-Syntax	58
	Ärger mit Strichpunkten	59
	Regeln zur Benennung von Variablen, Klassen etc.	60
	Java-Schlüsselwörter	61
	Kommentare im Java-Code	61
	Die Java-Klassenbibliothek	62
	Weniger Tippaufwand mit »import«	62
	Klassen aus der Standardbibliothek verwenden	64
2.3	Wiederholungsfragen	65
3	Variablenverwaltung	66
<hr/>		
3.1	Variablen	66
	Einführungsbeispiel	66
	Variablen deklarieren, initialisieren und verwenden	67
	Variablendeklaration ohne Typangabe (»Type Inference«)	68

3.2	Elementare Datentypen	69
	Ganze Zahlen	70
	Fließkommazahlen	71
	Rechnen mit »double«-Zahlen	72
	Boolesche Werte	73
	Zufallszahlen	73
	Typumwandlung (Casting)	74
	Modifizierer für die Variablendeklaration	75
3.3	Literale	77
	Boolesche Literale	77
	Ganze Zahlen	77
	Fließkommazahlen	79
3.4	Variablen im größeren Java-Kontext	79
	Gültigkeitsebenen	80
	Variablen für Objekte	81
	Wrapper-Klassen für elementare Datentypen	83
	Instanzvariablen (Fields)	85
3.5	Variablen einlesen und ausgeben	85
	Datenausgabe	86
	Dateneingabe	87
	Beispiel	87
3.6	Konstanten und Enums	89
	Konstanten	89
	Konstantenaufzählungen (Enums)	90
3.7	Wiederholungsfragen und Übungen	91
4	Operatoren	92
<hr/>		
4.1	Überblick	92
4.2	Details und Sonderfälle	94
	Zuweisungen	95

Mathematische Operatoren	95
Inkrement und Dekrement	97
Vergleiche	97
Boolesche Ausdrücke (verknüpfte Bedingungen)	98
Rechnen mit Bits	100
Sonstige Operatoren	101
4.3 Wiederholungsfragen	101
5 Verzweigungen und Schleifen	102
<hr/>	
5.1 »if«-Verzweigungen	103
Lieber ein Klammernpaar zu viel als eines zu wenig!	104
Klare Logik durch richtiges Einrücken	106
Beispiel: Schaltjahrtest	107
5.2 »if«-Kurzschreibweise (ternärer Operator)	108
5.3 »switch«-Verzweigungen	109
Beispiel: Tage pro Monat	110
»switch« in Java 12	111
5.4 »for«-Schleifen	113
Achtung, Falle!	114
Variablendeklaration innerhalb der Schleife	115
Beispiele	116
»for«-Schleifen für Fließkommazahlen	116
Verschachtelte Schleifen	118
5.5 »for-each«-Schleifen	119
»for« versus »for-each«	120
5.6 »while«- und »do-while«-Schleifen	121
»while«-Schleifen	121
»do-while«-Schleifen	122

5.7	»break« und »continue«	122
	break	122
	continue	123
	»break« und »continue« in verschachtelten Schleifen	123
	Endlosschleifen	124
5.8	Wiederholungsfragen und Übungen	124
6	Arrays	126
<hr/>		
6.1	Syntax	126
	Arrays initialisieren	126
	Zugriff auf Array-Elemente	127
	Mehrdimensionale Arrays	128
	Nichtrechteckige Arrays	128
	Interns	129
6.2	Mit Arrays arbeiten	131
	Methoden	131
	Arrays duplizieren	132
	Beispiel 1: Array initialisieren	132
	Beispiel 2: Minimum, Maximum und Mittelwert	133
6.3	Wiederholungsfragen	134
7	Zeichenketten	136
<hr/>		
7.1	Der Datentyp »char«	136
	Die »Character«-Klasse und ihre Methoden	137
7.2	Die »String«-Klasse	138
	»String«-Eigenschaften	139
	Zeichenketten vergleichen	140
	Zeichenketten korrekt ordnen und sortieren	141

	»String«-Methoden	142
	Die »join«-Methode	144
	Die Methoden »trim«, »strip« und »isBlank«	145
7.3	Formatierung und Konvertierung	145
	Formatierung	146
	Konvertierung von Zeichenketten in Zahlen	149
	Lokalisierung von Ein- und Ausgabe	149
7.4	Die »StringBuilder«-Klasse	151
7.5	Zeichensatzprobleme	152
	Quellcode	153
	Textausgabe im Terminal	154
7.6	Beispiele	154
	Groß- und Kleinbuchstaben zählen	155
	Pfad und Dateiname trennen	156
7.7	Wiederholungsfragen und Übungen	157
8	Datum und Uhrzeit	158
<hr/>		
8.1	Datum und Zeit seit Java 8	159
	»Machine Time Line« versus »Human Time Line«	159
	Überblick über die Klassen und Methoden	160
	Datum ermitteln, anzeigen und formatieren	162
	Schaltjahr-spezifische Daten ermitteln	163
	Uhrzeit ermitteln und anzeigen	164
	Daten und Zeiten einlesen (»parse«)	164
	Daten und Zeiten festlegen (»of«)	165
	Zeitspannen ermitteln und auswerten	165
	Rechnen mit Daten und Zeiten	166
	Rechenzeit messen (Instant und Duration)	167

8.2	Veraltete Datums- und Zeitklassen (Date, Calendar)	168
	Die »Date«-Klasse	168
	Formatierung mit »format« bzw. »printf«	170
	Formatierung mit der »SimpleDateFormat«-Klasse	170
	Die »Calendar«-Klasse	173
	Umwandlung von »Date« zu »LocalDate«	176
8.3	Wiederholungsfragen und Übungen	176
9	Methoden	177
<hr/>		
9.1	Einführung	178
	Syntaxregeln	179
	Statisch oder nichtstatisch?	180
9.2	Parameterliste	181
	Parameter verändern	181
	Finale Parameter	184
	Overloading	184
	Variable Parameterzahl	185
9.3	Rückgabewert und »return«	187
9.4	Rekursion	188
	Fakultät rekursiv berechnen	188
	Der Stack	189
9.5	Beispiele	190
	Array-Methoden: Minimum und Maximum ermitteln	190
	Wir spielen Lotto	191
9.6	Wiederholungsfragen und Übungen	194
10	Exceptions	197
<hr/>		
10.1	Exception-Klassen	198
	Die »Throwable«-Klasse	198

Die »Error«-Klassen	199
Die »RuntimeException«-Klassen	199
Gewöhnliche Exceptions	200
10.2 try-catch	200
»try-catch« für Ressourcen	201
Exception-Weitergabe	202
10.3 Fehleranfällige Methoden deklarieren (»throws«)	203
Selbst absichern oder die Absicherung delegieren?	204
10.4 Selbst Exceptions werfen (»throw«)	205
10.5 Beispiel	206
10.6 Wiederholungsfragen und Übungen	208
11 Klassen	209
<hr/>	
11.1 Top-Level-Klassen	210
Beispiel: Rechteck-Klasse	211
Gültigkeitsebenen (»public«, »private« und »protected«) ...	214
Klassenvariablen und statische Methoden	215
Konstruktor	218
this	219
Beispiel: Rechteck-Klasse mit Konstruktor	220
Destruktor, »finalize« und »close«	221
»get«- und »set«-Methoden (Getter/Setter)	222
Beispiel: Rechteck-Klasse mit Getter/Setter	223
11.2 Geschachtelte Klassen	225
Die Syntax geschachtelter Klassen	225
Geschachtelte Schnittstellen und Enums	227
11.3 Anonyme Klassen	227
Beispiel: »FilenameFilter«	228
Syntax	230
Variable Capture	231

11.4	Statische geschachtelte Klassen	232
11.5	Beispiel: Schachfigur Springer	233
	Aufgabenstellung	233
	Implementierung der »Springer«-Klasse	234
	Die Methode »ermittleZuege«	236
	Test	237
11.6	Beispiel: Bücher und Kapitel	238
	Chapter-Klasse	238
	Book-Klasse	239
	Test-Code	240
11.7	Wiederholungsfragen und Übungen	241
12	Vererbung und Schnittstellen	244
<hr/>		
12.1	Vererbung	245
	Methoden überschreiben	245
	Instanz- und Klassenvariablen verstecken	247
	super	248
	Konstruktor	248
	Finale Klassen und Methoden	249
	Abstrakte Klassen	250
	Generalisierung	251
	Polymorphie	252
	Upcasts und Downcasts	255
12.2	Die »Object«-Klasse	255
	Die Methode »clone«	256
	Die Methode »equals«	256
	Die Methode »finalize«	256
	Die Methode »getClass«	257
	Die Methode »hashCode«	257
	Die Methoden »notify«, »notifyAll« und »wait«	259

	Die Methode »toString«	259
12.3	Vererbungsbeispiel (Schachfiguren)	259
	Die abstrakte Klasse »Schachfigur«	259
	Die Klassen »Springer«, »Laeufer« und »Turm«	261
	Anwendung der Klassen	262
12.4	Schnittstellen	264
	Einführungsbeispiel	264
	Wichtige Schnittstellen in der Java-Standardbibliothek	265
	»interface«-Syntax	266
	Funktionale Schnittstellen und Defaultmethoden	267
	Die »implements«-Syntax	268
	Polymorphie bei Schnittstellen	269
	Abstrakte Klassen versus Schnittstellen	269
12.5	Schnittstellenbeispiel (geometrische Figuren)	269
	Rechteck- und Kreis-Klasse	270
	Anwendung der Klassen	271
12.6	Wiederholungsfragen und Übungen	272
13	Generische Klassen und Methoden	275
<hr/>		
13.1	Einführung	275
	Hello Generics World!	275
	Wrapper-Klassen	277
13.2	Deklaration generischer Klassen und Schnittstellen	277
	Typeinschränkungen	278
	Generische Schnittstellen und Vererbung	279
13.3	Deklaration generischer Methoden	279
13.4	Wildcards	280
	Wildcard-Parameter	281
	Wildcards mit Regeln	282

Upper Bounded Wildcards	283
Lower Bounded Wildcards	284
Arrays	284
13.5 Generics-Beispiel (Comparable)	285
Die »Geometrie«-Schnittstelle erweitern	286
Die »Kreis«-Klasse erweitern	286
Die »Rechteck«-Klasse erweitern	287
Die »Comparable«-Objekte sortieren	288
»Comparable« versus »Comparator«	288
13.6 Wiederholungsfragen und Übungen	290
14 Lambda-Ausdrücke	291
<hr/>	
14.1 Hello Lambda-World!	291
Ein Blick hinter die Kulissen	292
14.2 Lambda & Co.	293
Die Syntax von Lambda-Ausdrücken	293
»this« und »super«	295
Referenzen auf Methoden	295
Beispiel für Referenzen auf Methoden	297
Defaultmethoden	299
Generische Lambda-Schnittstellen	300
Beispiel: Datenselektion mit der »Predicate«-Schnittstelle	301
War das schon alles?	303
14.3 Wiederholungsfragen	303
15 Collections	305
<hr/>	
15.1 Einführung	305
Koordinatenpunkte eines Polygons speichern (»List«)	306

	Lottozahlen generieren (»Set«)	307
	Wörterbuch speichern (»Map«)	307
	Klassenüberblick	308
	Regeln, Tipps und Tricks	310
15.2	Die »Iterable«-Schnittstelle	312
	Die »forEach«-Methode	312
15.3	Die »Collection«-Schnittstelle	313
	Die »removeIf«- und »stream«-Methoden	314
15.4	Die »Set«-Schnittstelle	315
	Die »HashSet«-Klasse	315
	Die »LinkedHashSet«-Klasse	317
	Die »TreeSet«-Klasse	317
15.5	Die »List«-Schnittstelle	320
	Die »replaceAll«-Methode	321
	Die »ArrayList«-Klasse	322
	Die »LinkedList«-Klasse	322
15.6	Die »Stream«-Schnittstelle	323
	Stream-Beispiele	324
15.7	Die »Map«-Schnittstelle	327
	Die »HashMap«- und »LinkedHashMap«-Klassen	329
	Schleifen über Maps	329
15.8	Wiederholungsfragen und Übungen	331
16	Dateien und Verzeichnisse	332
<hr/>		
16.1	Klassen- und Schnittstellenüberblick	332
	Fehlerabsicherung	333
	Ressourcen schließen	334
16.2	Dateien und Verzeichnisse ergünden	334
	Besondere Verzeichnisse	334

Die »Path«-Schnittstelle	335
Testen, ob ein Verzeichnis bzw. eine Datei existiert	337
Eigenschaften einer Datei ermitteln	338
Liste der Dateien in einem Verzeichnis ermitteln	340
16.3 Dateien und Verzeichnisse bearbeiten	342
Beispiel	343
16.4 Textdateien lesen und schreiben	344
Textdateien schreiben	345
Textdateien auslesen	346
Andere Zeichensätze als UTF-8 verwenden	347
16.5 Wiederholungsaufgaben und Übungen	348
17 JavaFX	349
<hr/>	
17.1 Installation	349
17.2 Einführung	350
Manuell kompilieren	351
JavaFX-Programme mit IntelliJ entwickeln und ausführen	352
Ein erster Blick hinter die Kulissen	354
Der Scene Graph	355
17.3 Arbeiten mit Steuerelementen	356
Der Scene Graph des Beispielprogramms	358
Steuerelemente und Container erzeugen	358
Ereignisse	362
17.4 Grafikprogrammierung	364
Einführungsbeispiel	364
Den Zufall zeichnen lassen	367
Lissajous-Figuren zeichnen	368
17.5 Wiederholungsaufgaben und Übungen	371

18	Javadoc	372
<hr/>		
18.1	Javadoc-Syntax	372
	Beispiel	373
18.2	Das Javadoc-Kommando	376
18.3	Übung	376
19	Pakete, Bibliotheken und Module	377
<hr/>		
19.1	import	378
	Die »import«-Syntax	378
	Standard-Import für »java.lang«	379
	Statische Importe	379
19.2	Pakete	380
19.3	Bibliotheken	382
	Fertige Java-Bibliotheken nutzen	382
19.4	Module (»Jigsaw«)	384
	Beispiel	385
	Die Datei »module-info.java«	386
	Kompilieren und ausführen	387
	Module in der IntelliJ IDEA	388
19.5	Wiederholungsfragen	391
A	Crashkurs IntelliJ IDEA	392
<hr/>		
A.1	Benutzeroberfläche	392
	Effizient entwickeln	394
	Refactoring	395
	Ablenkungsfrei arbeiten	395

	Debugging	396
	Tastenkürzel	397
A.2	Projekte	398
	Code-Dateien hinzufügen	398
	Externe Bibliotheken nutzen	399
	Module	399
	Vorhandenen Code in ein IntelliJ-Projekt umwandeln	400
	Projektoptionen	401
A.3	Einstellungen	402
	Weniger Warnungen und Referenzen anzeigen	403
B	Lösungen	404
<hr/>		
B.1	Kapitel 1, »Hello World!«	404
B.2	Kapitel 2, »Java-Crashkurs«	406
B.3	Kapitel 3, »Variablenverwaltung«	408
B.4	Kapitel 4, »Operatoren«	410
B.5	Kapitel 5, »Verzweigungen und Schleifen«	411
B.6	Kapitel 6, »Arrays«	413
B.7	Kapitel 7, »Zeichenketten«	414
B.8	Kapitel 8, »Datum und Uhrzeit«	415
B.9	Kapitel 9, »Methoden«	415
B.10	Kapitel 10, »Exceptions«	417
B.11	Kapitel 11, »Klassen«	418
B.12	Kapitel 12, »Vererbung und Schnittstellen«	421
B.13	Kapitel 13, »Generische Klassen und Methoden«	423
B.14	Kapitel 14, »Lambda-Ausdrücke«	424
B.15	Kapitel 15, »Collections«	425
B.16	Kapitel 16, »Dateien und Verzeichnisse«	426