

luigi LO IACONO
stephan WIEFLING
michael SCHNEIDER

PROGRAMMIEREN TRAINIEREN

Mit über **130 Workouts**
in **Java** und **Python**

2. Auflage



Im Internet: **GitHub-Repository**
zum Buch

HANSER

Lo Iacono / Wiefling / Schneider
Programmieren trainieren



Blieben Sie auf dem Laufenden!

Unser **Computerbuch-Newsletter** informiert Sie monatlich über neue Bücher und Termine. Profitieren Sie auch von Gewinnspielen und exklusiven Leseproben. Gleich anmelden unter:

www.hanser-fachbuch.de/newsletter



Luigi Lo Iacono
Stephan Wiefling
Michael Schneider

Programmieren trainieren

Mit über 130 Workouts
in Java und Python

2., erweiterte Auflage

HANSER

Die Autoren:

Prof. Dr.-Ing. Luigi Lo Iacono, Bonn

Stephan Wiefeling, Köln

Michael Schneider, Daaden

Alle in diesem Buch enthaltenen Informationen, Verfahren und Darstellungen wurden nach bestem Wissen zusammengestellt und mit Sorgfalt getestet. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grund sind die im vorliegenden Buch enthaltenen Informationen mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autoren und Verlag übernehmen infolgedessen keine juristische Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieser Informationen – oder Teilen davon – entsteht. Ebenso übernehmen Autoren und Verlag keine Gewähr dafür, dass beschriebene Verfahren usw. frei von Schutzrechten Dritter sind.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Buch berechtigt deshalb auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung des Buches, oder Teilen daraus, vorbehalten. Kein Teil des Werkes darf ohne schriftliche Genehmigung des Verlages in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren) auch nicht für Zwecke der Unterrichtsgestaltung reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

© 2020 Carl Hanser Verlag München, www.hanser-fachbuch.de

Lektorat: Sylvia Hasselbach

Copy editing: Walter Saumweber, Ratingen

Herstellung: Irene Weilhart

Layout: le-tex publishing services GmbH

Umschlagdesign: Marc Müller-Bremer, www.rebranding.de, München

Umschlagrealisation: Max Kostopoulos

Datenbelichtung, Druck und Bindung: Kösel, Krugzell

Printed in Germany

Print-ISBN: 978-3-446-45911-3

E-Book-ISBN: 978-3-446-46057-7

E-Pub-ISBN: 978-3-446-46494-0

Inhalt

Vorwort	XVII
Danksagung	XIX
1 Einleitung.....	1
1.1 Wozu sollte ich programmieren lernen?	1
1.2 Wie kann mir dieses Buch dabei helfen?	2
1.3 Was muss ich mitbringen?	2
1.4 Wie geht das vonstatten?	3
1.5 Was muss ich sonst noch wissen?	5
2 Einführung in die Programmierung.....	7
2.1 Warmup	7
2.2 Workout	11
W.2.1 Three-Two-One – Mein erstes Programm.....	11
W.2.2 Weihnachtsbaum	12
W.2.3 Perlenkette	14
W.2.4 Die erste Zeichnung	15
W.2.5 Raupe Allzeitappetit	17
W.2.6 Klötzchen-Kunst.....	18
W.2.7 Nachteule	19
W.2.8 Ghettoblaster	20
W.2.9 Hallo Bello!	22
W.2.10 Haus.....	23
W.2.11 Daumen	24
3 Variablen, Datentypen, Operatoren und Ausdrücke	25
3.1 Warmup	25
3.2 Workout	28
W.3.1 Einfache Rechenaufgaben	28
W.3.2 Perlenkette 2.0	29
W.3.3 Blutalkoholkonzentration.....	30
W.3.4 Stoffwechselrate	32
W.3.5 Baumstammvolumen	34
W.3.6 Körperoberfläche	35

W.3.7	RGB nach CMYK	36
W.3.8	Tic-Tac-Toe-Spielfeld	38
W.3.9	Haus mit Garage	39
W.3.10	Fußballtor	40
4	Kontrollstrukturen	43
4.1	Warmup	43
4.2	Workout	47
W.4.1	Maximum bestimmen	47
W.4.2	Summe berechnen	48
W.4.3	Tippspiel	49
W.4.4	PIN-Code-Generator	50
W.4.5	Dominosteine	51
W.4.6	Radialer Farbverlauf	52
W.4.7	Ladevorgang-Rädchen	53
W.4.8	Windrad	55
W.4.9	Rotierte Dreiecke	56
W.4.10	Moderne Kunst	57
W.4.11	Schachbrett	59
W.4.12	Sinuskurve	60
W.4.13	Zahlen-Palindrom	61
W.4.14	Interaktiver Button	62
W.4.15	Ebbe und Flut berechnen	64
W.4.16	Titschender Ball	65
5	Funktionen	67
5.1	Warmup	67
5.2	Workout	69
W.5.1	Endliches Produkt	69
W.5.2	Fakultät	70
W.5.3	Konfektionsgröße	71
W.5.4	Schaltjahr Prüfung	72
W.5.5	Literzahlen umwandeln	73
W.5.6	LKW-Maut	74
W.5.7	Analoger Uhrzeiger	75
W.5.8	Körperoberfläche 2.0	76
W.5.9	Sportwetten	77
W.5.10	GPS-Luftlinie	79
W.5.11	IBAN-Generator	81
W.5.12	Sanduhr	82
W.5.13	Der faire Würfel	83
W.5.14	Quadrat mit Kreisausschnitten	84
W.5.15	Wurfparabel	86
W.5.16	Bogenschießen-Spiel	88
W.5.17	Mondphase berechnen	90
W.5.18	Tortendiagramm	92
W.5.19	Pendelanimation	94

6	Arrays	97
6.1	Warmup	97
6.2	Workout	100
W.6.1	Wochentag	100
W.6.2	Tankfüllung	102
W.6.3	Rückwärtsausgabe	103
W.6.4	Bildvergrößerung	104
W.6.5	Partnervermittlung	105
W.6.6	Sitzplatzreservierung	106
W.6.7	Platztausch	107
W.6.8	Minimale Distanz	108
W.6.9	Morsecode	109
W.6.10	Endlose Animation	110
W.6.11	Spiegeln	111
W.6.12	Reflexion	112
W.6.13	Greenscreen	114
W.6.14	Bild umdrehen und invertieren	115
W.6.15	Bild mit Schatten	116
W.6.16	Bild rotieren	117
W.6.17	Bildverkleinerung	118
W.6.18	Minimale Punktdistanz	119
W.6.19	Glatte Kurven	121
W.6.20	Bildausschnitt	123
W.6.21	Bild mit Rahmen	124
W.6.22	Memory-Spielfeldgenerator	125
W.6.23	Sudoku-Check	127
W.6.24	Medianfilter	128
W.6.25	Geldautomat	130
W.6.26	Postleitzahlen visualisieren	131
W.6.27	Dreiecksfilter	133
7	Strings und Stringverarbeitung	135
7.1	Warmup	135
7.2	Workout	137
W.7.1	String-Kompression	137
W.7.2	Split-Funktion	138
W.7.3	Geldschein-Blütencheck	139
W.7.4	Starkes Passwort	140
W.7.5	E-Mail-Check	141
W.7.6	Prüfen auf eine korrekte Klammerung	142
W.7.7	Sternchenmuster	143
W.7.8	URL-Encoding	144
W.7.9	Telefonbuch bearbeiten	145
W.7.10	Webserver-Antwort verarbeiten	147
W.7.11	IMDB-Einträge verarbeiten	149
W.7.12	Geheimsprache	150
W.7.13	Ähnlich klingende Wörter	151

W.7.14	Textrahmen	153
W.7.15	JSON-Array.....	154
W.7.16	Barcode-Generator	155
W.7.17	Kennzeichenverarbeitung	157
8	Objektorientierung	159
8.1	Warmup	159
8.2	Workout	162
W.8.1	Schrittzähler	162
W.8.2	Body-Mass-Index.....	164
W.8.3	Songtextsuche	166
W.8.4	Passwortklasse	167
W.8.5	Kopffitness	169
W.8.6	Fernbedienung	170
W.8.7	Stoppuhr	171
W.8.8	Druckerwarteschlange	172
W.8.9	Tic Tac Toe.....	174
W.8.10	Zwischenablage	176
W.8.11	Temperaturgraph.....	177
W.8.12	Ambient Light	179
W.8.13	Verschlüsselung	183
W.8.14	Mastermind	185
W.8.15	Parteistimmen	187
9	Referenzdatentypen	189
9.1	Warmup	189
9.2	Workout	191
W.9.1	Kreis-Klasse	191
W.9.2	Mathematischer Bruch	192
W.9.3	Highscore-Liste.....	193
W.9.4	Adressbuch.....	194
W.9.5	Digitaler Bilderrahmen	196
W.9.6	Musikalbenanwendung	197
W.9.7	Koch-Website	199
W.9.8	Hotelzimmerverwaltung	201
W.9.9	Flughafen-Check-in	203
W.9.10	Polygonzug.....	205
W.9.11	Twitterwall	207
W.9.12	Fototagebuch	209
W.9.13	Partygäste	211
W.9.14	Raumbelegung	213
10	Vererbung	217
10.1	Warmup	217
10.2	Workout	219
W.10.1	Online-Shop	219
W.10.2	Gewässer	221

W.10.3 To-do-Liste	222
W.10.4 Lampen	224
W.10.5 Meeting-Protokoll	225
W.10.6 E-Book	227
W.10.7 Zoo	229
W.10.8 Audioeffekt-Player	230
W.10.9 Fahrtenbuch	231
W.10.10 Webseitengenerator	232
A Lösungen in Java	235
A.1 Download und Verwendung der elektronischen Lösungen	235
A.1.1 Download von GitHub	235
A.1.2 Öffnen der Programme	236
A.1.3 Tipp: Debugger	237
A.2 Einführung in die Programmierung	238
A.2.1 Three-Two-One – Mein erstes Programm	238
A.2.2 Weihnachtsbaum	238
A.2.3 Perlenkette	238
A.2.4 Die erste Zeichnung	239
A.2.5 Nachteule	239
A.2.6 Raupe Allzeitappetit	240
A.2.7 Klötzchen-Kunst	240
A.2.8 Ghettoblaster	241
A.2.9 Hallo Bello!	242
A.2.10 Haus	244
A.2.11 Daumen	244
A.3 Variablen, Datentypen, Operatoren und Ausdrücke	246
A.3.1 Einfache Rechenaufgaben	246
A.3.2 Perlenkette 2.0	247
A.3.3 Blutalkoholkonzentration	247
A.3.4 Stoffwechselrate	248
A.3.5 Baumstammvolumen	248
A.3.6 Körperoberfläche	248
A.3.7 RGB nach CMYK	249
A.3.8 Tic-Tac-Toe-Spielfeld	249
A.3.9 Haus mit Garage	250
A.3.10 Fußballtor	251
A.4 Kontrollstrukturen	253
A.4.1 Maximum bestimmen	253
A.4.2 Summe berechnen	253
A.4.3 Tippspiel	254
A.4.4 PIN-Code-Generator	254
A.4.5 Dominosteine	255
A.4.6 Radialer Farbverlauf	255
A.4.7 Ladevorgang-Rädchen	256
A.4.8 Windrad	256
A.4.9 Rotierte Dreiecke	256

A.4.10	Moderne Kunst	257
A.4.11	Schachbrett	258
A.4.12	Sinuskurve	258
A.4.13	Zahlen-Palindrom	259
A.4.14	Interaktiver Button	260
A.4.15	Ebbe und Flut berechnen	261
A.4.16	Titschender Ball	262
A.5	Funktionen	264
A.5.1	Endliches Produkt	264
A.5.2	Fakultät	264
A.5.3	Konfektionsgröße	265
A.5.4	Schaltjahr Prüfung	266
A.5.5	Literzahlen umwandeln	266
A.5.6	LKW-Maut	267
A.5.7	Analoger Uhrzeiger	268
A.5.8	Körperoberfläche	269
A.5.9	Sportwetten	270
A.5.10	GPS-Luftlinie	271
A.5.11	IBAN-Generator	272
A.5.12	Sanduhr	273
A.5.13	Der faire Würfel	274
A.5.14	Quadrat mit Kreisausschnitten	276
A.5.15	Wurfparabel	277
A.5.16	Bogenschießen-Spiel	278
A.5.17	Mondphase berechnen	282
A.5.18	Tortendiagramm	283
A.5.19	Pendelanimation	285
A.6	Arrays	287
A.6.1	Wochentag	287
A.6.2	Tankfüllung	288
A.6.3	Rückwärtsausgabe	289
A.6.4	Bildvergrößerung	290
A.6.5	Partnervermittlung	291
A.6.6	Sitzplatzreservierung	292
A.6.7	Platztausch	294
A.6.8	Bestimmung minimale Distanz	294
A.6.9	Morsecode	295
A.6.10	Endlose Animation	297
A.6.11	Spiegeln	298
A.6.12	Reflexion	299
A.6.13	Greenscreen	301
A.6.14	Bild umdrehen und invertieren	302
A.6.15	Bild mit Schatten	303
A.6.16	Bild rotieren	305
A.6.17	Bildverkleinerung	306
A.6.18	Minimale Punktdistanz	308
A.6.19	Glatte Kurven	309

A.6.20	Bildausschnitt	310
A.6.21	Bild mit Rahmen	312
A.6.22	Memory-Spielfeldgenerator	313
A.6.23	Sudoku-Check	315
A.6.24	Medianfilter	318
A.6.25	Geldautomat	319
A.6.26	Postleitzahlen visualisieren	320
A.6.27	Dreiecksfilter	321
A.7	Strings und Stringverarbeitung	324
A.7.1	String-Kompression	324
A.7.2	Split-Funktion	325
A.7.3	Geldschein-Blütencheck	326
A.7.4	Starkes Passwort	329
A.7.5	E-Mail-Check	330
A.7.6	Prüfen auf eine korrekte Klammerung	331
A.7.7	Sternchenmuster	332
A.7.8	URL-Encoding	333
A.7.9	Telefonbuch bearbeiten	335
A.7.10	Webserver-Antwort verarbeiten	337
A.7.11	IMDB-Einträge verarbeiten	339
A.7.12	Geheimsprache	340
A.7.13	Ähnlich klingende Wörter	340
A.7.14	Textrahmen	342
A.7.15	JSON-Array	343
A.7.16	Barcode-Generator	344
A.7.17	Kennzeichenverarbeitung	348
A.8	Objektorientierung	350
A.8.1	Schrittzähler	350
A.8.2	Body-Mass-Index	351
A.8.3	Songtextsuche	351
A.8.4	Passwortklasse	352
A.8.5	Kopffitness	355
A.8.6	Fernbedienung	355
A.8.7	Stoppuhr	357
A.8.8	Druckerwarteschlange	359
A.8.9	Tic Tac Toe	360
A.8.10	Zwischenablage	362
A.8.11	Temperaturgraph	364
A.8.12	Ambient Light	367
A.8.13	Verschlüsselung	369
A.8.14	Mastermind	371
A.8.15	Parteistimmen	374
A.9	Referenzdatentypen	376
A.9.1	Kreis-Klasse	376
A.9.2	Mathematischer Bruch	377
A.9.3	Highscore-Liste	378
A.9.4	Adressbuch	380

A.9.5	Digitaler Bilderrahmen	384
A.9.6	Musikalbenanwendung	386
A.9.7	Koch-Website	388
A.9.8	Hotelzimmerverwaltung	391
A.9.9	Flughafen-Check-in	393
A.9.10	Polygonzug	396
A.9.11	Twitterwall	398
A.9.12	Fototagebuch	399
A.9.13	Partygäste	402
A.9.14	Raumbelegung	404
A.10	Vererbung	409
A.10.1	Online-Shop	409
A.10.2	Gewässer	412
A.10.3	To-do-Liste	415
A.10.4	Lampen	419
A.10.5	Meeting-Protokoll	420
A.10.6	E-Book	423
A.10.7	Zoo	428
A.10.8	Audioeffekt-Player	430
A.10.9	Fahrtenbuch	433
A.10.10	Webseitengenerator	435
B	Lösungen in Python	441
B.1	Download und Verwendung der elektronischen Lösungen	441
B.1.1	Download von GitHub	441
B.1.2	Öffnen der Programme	441
B.2	Einführung in die Programmierung	443
B.2.1	Three-Two-One - Mein erstes Programm	443
B.2.2	Weihnachtsbaum	443
B.2.3	Perlenkette	443
B.2.4	Die erste Zeichnung	444
B.2.5	Nachteule	444
B.2.6	Raupe Allzeitappetit	445
B.2.7	Klötzchenkunst	445
B.2.8	Ghettoblaster	446
B.2.9	Hallo Bello!	448
B.2.10	Haus	449
B.2.11	Daumen	449
B.3	Variablen, Datentypen, Operatoren und Ausdrücke	451
B.3.1	Einfache Rechenaufgaben	451
B.3.2	Perlenkette 2.0	452
B.3.3	Blutalkoholkonzentration	452
B.3.4	Stoffwechselrate	453
B.3.5	Baumstammvolumen	453
B.3.6	Körperoberfläche	453
B.3.7	RGB nach CMYK	454
B.3.8	Tic-Tac-Toe-Spielfeld	454

B.3.9	Haus mit Garage	455
B.3.10	Fußballtor	456
B.4	Kontrollstrukturen	458
B.4.1	Maximum bestimmen	458
B.4.2	Summe berechnen	458
B.4.3	Tippspiel	458
B.4.4	PIN-Code-Generator	459
B.4.5	Dominosteine	459
B.4.6	Radialer Farbverlauf	460
B.4.7	Ladevorgang-Rädchen	460
B.4.8	Windrad	461
B.4.9	Rotierte Dreiecke	461
B.4.10	Moderne Kunst	461
B.4.11	Schachbrett	462
B.4.12	Sinuskurve	463
B.4.13	Zahlen-Palindrom	464
B.4.14	Interaktiver Button	465
B.4.15	Ebbe und Flut berechnen	465
B.4.16	Titschender Ball	466
B.5	Funktionen	468
B.5.1	Endliches Produkt	468
B.5.2	Fakultät	468
B.5.3	Konfektionsgröße	469
B.5.4	Schaltjahr Prüfung	470
B.5.5	Literzahlen umwandeln	470
B.5.6	LKW-Maut	471
B.5.7	Analoger Uhrzeiger	472
B.5.8	Körperoberfläche	473
B.5.9	Sportwetten	473
B.5.10	GPS-Luftlinie	474
B.5.11	IBAN-Generator	475
B.5.12	Sanduhr	476
B.5.13	Der faire Würfel	477
B.5.14	Quadrat mit Kreisausschnitten	478
B.5.15	Wurfparabel	479
B.5.16	Bogenschießen-Spiel	480
B.5.17	Mondphase berechnen	484
B.5.18	Tortendiagramm	485
B.5.19	Pendelanimation	486
B.6	Arrays	488
B.6.1	Wochentag	488
B.6.2	Tankfüllung	489
B.6.3	Rückwärtsausgabe	490
B.6.4	Bildvergrößerung	490
B.6.5	Partnervermittlung	492
B.6.6	Sitzplatzreservierung	493
B.6.7	Platztausch	494

B.6.8	Bestimmung minimale Distanz	495
B.6.9	Morsecode	496
B.6.10	Endlose Animation.....	497
B.6.11	Spiegeln.....	498
B.6.12	Reflexion.....	499
B.6.13	Greenscreen.....	500
B.6.14	Bild umdrehen und invertieren	501
B.6.15	Bild mit Schatten	503
B.6.16	Bild rotieren	504
B.6.17	Bildverkleinerung	506
B.6.18	Minimale Punktdistanz	507
B.6.19	Glatte Kurven	508
B.6.20	Bildausschnitt	510
B.6.21	Bild mit Rahmen	511
B.6.22	Memory-Spielfeldgenerator	512
B.6.23	Sudoku-Check	514
B.6.24	Medianfilter	516
B.6.25	Geldautomat	517
B.6.26	Postleitzahlen visualisieren	518
B.6.27	Dreiecksfilter	520
B.7	Strings und Stringverarbeitung	522
B.7.1	String-Kompression.....	522
B.7.2	Split-Funktion	522
B.7.3	Geldschein-Blütencheck	523
B.7.4	Starkes Passwort	525
B.7.5	E-Mail-Check	526
B.7.6	Prüfen auf eine korrekte Klammerung	527
B.7.7	Sternchenmuster	528
B.7.8	URL-Encoding	529
B.7.9	Telefonbuch bearbeiten	529
B.7.10	Webserver-Antwort verarbeiten.....	531
B.7.11	IMDB-Einträge verarbeiten	532
B.7.12	Geheimsprache.....	533
B.7.13	Ähnlich klingende Wörter	534
B.7.14	Textrahmen	535
B.7.15	JSON-Array.....	536
B.7.16	Barcode-Generator	537
B.7.17	Kennzeichenverarbeitung	540
B.8	Objektorientierung	542
B.8.1	Schrittzähler	542
B.8.2	Body-Mass-Index.....	542
B.8.3	Songtextsuche	543
B.8.4	Passwortklasse.....	544
B.8.5	Kopffitness	546
B.8.6	Fernbedienung	547
B.8.7	Stoppuhr	548
B.8.8	Druckerwarteschlange	549

B.8.9	Tic Tac Toe.....	550
B.8.10	Zwischenablage	552
B.8.11	Temperaturgraph.....	554
B.8.12	Ambient Light	556
B.8.13	Verschlüsselung	559
B.8.14	Mastermind	562
B.8.15	Parteistimmen	564
B.9	Referenzdatentypen	566
B.9.1	Kreis-Klasse	566
B.9.2	Mathematischer Bruch	567
B.9.3	Highscore-Liste.....	568
B.9.4	Adressbuch	569
B.9.5	Digitaler Bilderrahmen	572
B.9.6	Musikalbenanwendung	574
B.9.7	Koch-Website	576
B.9.8	Hotelzimmerverwaltung	578
B.9.9	Flughafen-Check-in	580
B.9.10	Polygonzug	581
B.9.11	Twitterwall	583
B.9.12	Fototagebuch	584
B.9.13	Partygäste	586
B.9.14	Raumbelegung	589
B.10	Vererbung	592
B.10.1	Online-Shop	592
B.10.2	Gewässer	594
B.10.3	To-do-Liste	597
B.10.4	Lampen.....	599
B.10.5	Meeting-Protokoll.....	601
B.10.6	E-Book	603
B.10.7	Zoo	606
B.10.8	Audioeffekt-Player	609
B.10.9	Fahrtenbuch	611
B.10.10	Webseitengenerator	613
C	Installation Processing	619
C.1	Einleitung	619
C.2	Windows	619
C.3	macOS.....	620
C.4	Linux.....	621
C.5	Aktivierung des Python Mode.....	622
D	Howto: Buch-Aufgaben ohne Processing lösen	625
D.1	Java.....	625
D.2	Python.....	625
D.3	Fazit.....	628



Vorwort

„Jede hinreichend fortschrittliche Technologie ist von Magie nicht zu unterscheiden.“

Arthur C. Clarke

Wir sind Programmierer. Wir sind Magier. Das MIT ist unser Hogwarts, der Google Campus ist unsere Unsichtbare Universität, Cupertino ist unser Narnia. Steve Jobs ist unser David Copperfield, Larry Page und Sergey Brin sind unsere Ehrlich Brothers und Frank Thelen ist immerhin vielleicht noch sowas wie unser Vincent Raven. Wir sind Siegfried und Roy und aus Versehen programmierte Endlosschleifen sind unsere weißen Tig. . . strapazieren wir die Allegorie mal nicht über. Jedenfalls: Wir sind Magier.

Oder wenigstens wirken wir für unser Umfeld so. Der Onkel dritten Grades, der im Atomkraftwerk arbeitet, würde uns selbst dann anrufen, wenn AssetWise mal hakt, weil wir eben Informatiker sind und uns dementsprechend mit allem auskennen, was irgendwie mit Computern zusammenhängt. Dabei ist es auch egal, wie komplex oder unterkomplex die Aufgabe ist. Wir werden angerufen, wenn ein Teilchenbeschleuniger angesteuert werden muss, aber auch, wenn es im Fachgeschäft für Strickzubehör „Woll im Leben“ einer Freundin des Freundes deiner Tante väterlicherseits in der alten Fußgängerzone der Kleinstadt, in der du geboren wurdest, nicht mehr ganz so gut läuft und sie jetzt „mal eben“ einen Shop braucht, um den Globalisierungseffekt besser für sich zu nutzen und das Wollgeschäft im Sturm zu erobern. Gestrickt wird ja wohl überall und sie ist sogar bereit, dir für die 3 Wochen Arbeit noch 100 € in die Hand zu drücken. Dafür müsstest du dann aber auch die nächsten 3 Jahre zu jeder Tages- und Nachtzeit für Support zur Verfügung stehen.

Doch wir steuern nicht nur die Stromversorgung und das weltweite Woll-Business. Wir halten Banken am Laufen, das Transportwesen und die Kommunikation, ohne uns läuft gar nichts mehr heutzutage. Wir können Welten erschaffen und wir können sie auch zerstören, je nachdem, ob wir Harry Potter oder Lord Voldemort sein wollen.

Welchen Weg du einschlagen willst – und jeder, der schon mal programmiert hat und behauptet, niemals auf die dunkle Seite geschaut zu haben, lügt – entscheidest du selbst und auf dem Weg zu deiner Entscheidung ist dieses Buch dein Hogwarts Express und du musst nicht mal gegen eine Mauer rennen, um hinein zu gelangen. Du hast die erste Seite aufgeschlagen und das Vorwort gelesen und die Richtung, in der du weiterblätterst, ist deine rote und deine blaue Pille.

Schlag es wieder zu – dann endet die Geschichte hier, du wachst auf in deinem Bett und glaubst, was immer du glauben möchtest. Blätter weiter, bleib im Wunderland und das Buch zeigt dir, wie tief der Kaninchenbau geht. Nerd today, boss tomorrow.

Patrick Stenzel (@rock_galore), im Januar 2020

■ Vorwort zur ersten Auflage

Nerds sind in. Diese liebenswerten Zeitgenossen mit dem vielen Spezialwissen und den kindlichen Vorlieben für Superhelden werden lange nicht mehr nur komisch beäugt. Im Gegenteil. Sie selbst sind nunmehr Stars in vielen Fernsehserien, und ihr modischer Stil ist allgemein akzeptiert. Diese Entwicklung kommt auch der Programmierung zugute. Lange Zeit galt diese Fertigkeit als ein Gebiet, das den Nerds vorbehalten ist. Dem ist nicht so! Es muss nur der Mut aufgebracht werden, sich damit auseinanderzusetzen. Dann wird schnell klar, was mit der Programmierung alles umgesetzt werden kann. Die Bandbreite ist groß und wird durch aktuelle Trends stetig befeuert. Insbesondere durch die Digitalisierung und Vernetzung vieler Alltagsgegenstände finden sich Softwareprogramme vermehrt jenseits gängiger Anwendungsfälle im betrieblichen Kontext von Unternehmen wieder. Also, keine Scheu und ran ans Programmieren!

Mir selbst bereitet das Programmieren viel Freude. Zudem ist es mir eine Herzensangelegenheit, mein Programmier-Knowhow und meine Erfahrung an andere weiterzugeben. Ich weiß aus vielen Schulungen sehr genau, was es für Hürden und Stolpersteine beim Programmieren lernen gibt und wie diesen zu begegnen ist. **Gutem Trainingsmaterial kommt dabei eine zentrale Rolle zu.**

Die Autoren Lo Iacono, Wiefling und Schneider schließen hier eine wichtige Lücke. Sie versorgen dich mit vielen Trainingsaufgaben, die dir helfen werden, die wesentlichen Programmierkonzepte wirklich zu verstehen. Und mehr noch. Du kannst und solltest so lange mit den vielen Aufgaben trainieren, bis der Groschen tatsächlich gefallen ist. Das ist wichtig. Denn erst dann wirst Du in der Lage sein, mit dem erlernten Handwerkszeug auch selbstständig Entwicklungsaufgaben bewältigen und lösen zu können. Genau da sollst du hin. Viele Lehrformate gehen hier nicht weit genug. Die falsche Annahme ist dabei häufig, dass ein Beispiel und eine Übungsaufgabe zum Verständnis ausreichen. Weit gefehlt. Es fängt schon damit an, dass nicht jeder mit dem gegebenen Beispiel oder der gestellten Übungsaufgabe etwas anfangen kann. Hier schafft das vorliegende Buch Abhilfe, und es gehört damit in die „Einstieg in die Programmierung“-Ecke deines Bücherregals.

Dirk Louis, Januar 2018

Danksagung

Ein Buchprojekt ist harte Arbeit. Ohne die Unterstützung vieler helfender Hände geht es nicht. Wir können uns gar nicht genug bei euch allen bedanken, versuchen es aber dennoch, so gut wir können.

Zeit ist wohl das Kostbarste, was wir haben. Darum bin ich umso dankbarer, dass meine liebe Familie mir diese für derartige und andere Projekte einräumt. Danke Barbara, Giuliana, Antonio und Fabio.

Danke an Brigitte und Frank, Christian, Stephan, Viet, Peter und allen, die mich bei der Arbeit an diesem Buch unterstützt haben. Außerdem danke ich allen Förderern, besonders meinem damaligen Informatiklehrer Herr Schepanek.

Ich danke meiner Frau Sabrina und meinen Kindern Juli und Lana für ihre Unterstützung und die mir für dieses Projekt eingeräumte Zeit. Außerdem möchte ich mich bei den vielen Kollegen sowie Auszubildenden bedanken, die mich auf meinem beruflichen Weg begleitet haben.

Gemeinsam möchten wir uns bei Christian Ullenboom bedanken. Er hat das Buch kritisch durchgearbeitet und uns viele wertvolle Anmerkungen dazu gegeben. Gleiches gilt für Dirk Louis, der uns zudem freundlicherweise das Vorwort zur Erstauflage des Buches geschrieben hat. Bei Patrick Stenzel möchten wir uns für das Vorwort zu dieser zweiten Auflage bedanken. Dem Hanser Verlag und insbesondere Frau Sylvia Hasselbach möchten wir für die viele Geduld mit uns und die ungebrochene Unterstützung bedanken. Ein derartiger Rückhalt ist unverzichtbar.

Außerdem bedanken wir uns bei Lars Wildeshaus und Jens Schönbohm, die uns auf kleine Fehler in der Erstauflage aufmerksam gemacht haben, welche wir in dieser Auflage korrigieren konnten.

Schließlich wollen wir uns an dieser Stelle bei den vielen Freiwilligen da draußen bedanken, die ihre wertvolle Zeit dafür aufwenden, um der Allgemeinheit viele nützliche Dinge kosten- und diskriminierungsfrei zur Verfügung zu stellen. Unser Buch bedient sich viel aus der Public Domain, wofür wir uns gerne durch Benennung der wesentlichen Bausteine bedanken wollen. Zunächst sind hierzu die beiden zugrunde gelegten Programmiersprachen zu nennen. **Java** (<https://www.java.com/de/>) wird von Oracle Corporation und **Python** (<https://www.python.org/>) von der Python Software Foundation bereitgestellt. Beide gehören aktuell zu den am meisten eingesetzten Sprachen und können auf vielfältigste Weise verwendet werden. Um die Hürden gerade für (fachfremde) Einsteiger in die Programmierung weitestgehend zu eliminieren, stellt die Processing Foundation die gleichnamige

Entwicklungsumgebung zur Verfügung (<https://processing.org/>). **Processing** basiert dabei von Hause aus auf Java. Im Laufe der Zeit sind weitere Programmiersprachen hinzugekommen, darunter neben JavaScript auch Python (<http://py.processing.org/>). Wir verwenden im Buch zudem **Piktogramme**, um die Aufgaben durch kleine Icons visuell zu unterstreichen. Iconify.it stellt unter <https://iconify.it/downloads/iconify-650-free-icons/> eine Sammlung von 650 freien Glyphicons bereit, aus der wir uns hierzu bedient haben. Schließlich verwenden wir in einigen Programmieraufgaben Bilder, die durch das Programm verarbeitet werden sollen. Hier haben wir auf die Public-Domain-Cliparts von clker.com (<http://clker.com/>) zurückgegriffen.

Last, but not least wollen wir uns **bei dir bedanken**. Wir freuen uns, dass wir dein Interesse geweckt und es schon mal bis in deine Hände geschafft haben. Jetzt bleibt uns nur noch, dir beim Trainieren viel Erfolg zu wünschen.

Luigi Lo Iacono, Stephan Wiefeling und Michael Schneider
April 2020

1

Einleitung

■ 1.1 Wozu sollte ich programmieren lernen?

Weil Du es kannst und weil die Programmierung **das Werkzeug des 21. Jahrhunderts** ist. Die Bundeskanzlerin Frau Angela Merkel hat in einem Interview mit YouTubern das Programmieren auf eine Stufe mit den Grundfertigkeiten Lesen, Schreiben und Rechnen gestellt (<https://youtu.be/Uq2zIzscPgY?t=12m18s>). Programmieren ist lange nicht mehr nur etwas für Experten, die das studiert haben. Durch den Einzug des Digitalen in alle Branchen und den Alltag können viele andere als nur Informatiker von der Programmierung profitieren und damit ihre Ideen erproben und verwirklichen. Beispiele kannst du unzählige finden. Lass' uns hier nur einige zur Verdeutlichung kurz anreißen. Dir fallen dann bestimmt selbst viele weitere Beispiele ein.

Angenommen, du bist **Künstler** und hast bisher mit den klassischen Materialien und Techniken deiner Disziplin gearbeitet. Für deine neueste Projektidee möchtest du mit regelmäßigen Formen und Farben experimentieren, wie es z. B. Sol LeWitt in seinem künstlerischen Schaffen getan hat (https://de.wikipedia.org/wiki/Sol_LeWitt). Das erfordert viel Fleiß, Geduld und Präzision. Da du deine Zeit lieber damit verbringen möchtest, an spannenden neuen Konstruktionen und deren Wirkung zu experimentieren, anstatt diese in langwierigen und teils monotonen Arbeitsschritten erst erstellen zu müssen, wünschst du dir einen Automatismus dafür, der das für dich erledigt. Dies kann ein eigens geschriebenes Computerprogramm leisten. Ist ein solches geschrieben, liegen die Vorteile auf der Hand. Veränderungen an den Farben, der Größe sowie Anordnungen der Formen usw. sind umgehend gemacht. Auch das Ausgabeformat kann leicht angepasst werden, um das Kunstwerk in vielfältiger Art und Weise zu drucken oder aus einem Rohling zu fräsen. Pioniere der computergenerierten Kunst sind z. B. Manfred Mohr, Joseph Nechvatal, Olga Kisseleva und John Lansdown.

Als **Veranstaltungstechniker** sieht man sich heute immer stärker mit Anforderungen von Kunden konfrontiert, die nach noch nicht dagewesenen Hinguckern verlangen. Hierfür gibt es naturgemäß keine fertigen Lösungen, die man aus dem Regal ziehen kann. Somit siehst du dich auf der einen Seite immer mit neuen spannenden Entwicklungsaufgaben konfrontiert, musst dafür aber auf der anderen Seite adäquate Lösungen entwickeln. Diese bedingen eigentlich immer auch Software, die es zu programmieren gilt.

Im letzten fiktiven Szenario wollen wir ins **Internet der Dinge** abtauchen. Mit diesem Schlagwort wird der allgemeine Trend bezeichnet, mit dem die Digitalisierung und die Vernetzung

im Gewand des Internets stetig in Gegenstände des alltäglichen Gebrauchs diffundieren. Der smart gewordene Fernseher ist ein Paradebeispiel hierfür. Einige neue Anwendungen findest du toll, willst aber noch nicht in neue Produkte investieren. Die alten tun es ja noch. So findest du es z. B. praktisch, im Supermarkt einen Blick in deinen Kühlschrank werfen zu können, um zu sehen, ob es genügend Frühstückseier fürs Wochenende gibt. Der Kühlschrank ist schnell für diesen Anwendungsfall erweitert. Mit einer batteriebetriebenen Kamera, einem LED-Licht und etwas Programmierung kannst du bald via Smartphone-App in deinen Kühlschrank gucken.

Das soll zeigen, was dir alles an Möglichkeiten offen steht, wenn du die Programmierung als ein Werkzeug verstehst und dich dessen bemächtigst.

■ 1.2 Wie kann mir dieses Buch dabei helfen?

Vor den Erfolg haben die Götter allerdings den Schweiß gesetzt. Diese Tatsache hat der griechische Dichter und Geschichtsschreiber *Hesiod* bereits vor langer Zeit festgestellt und dann so zutreffend formuliert. Dieser Ausspruch trifft unseres Erachtens kaum besser auf etwas zu als die Programmierung. Es gehört eine ordentliche Portion Arbeit dazu, bis der Groschen fällt und man die wesentlichen Programmierkonzepte verstanden hat. Erst dann wird man in der Lage sein, Aufgabenstellungen jeglicher Couleur selbstständig angehen und erfolgreich bewältigen zu können. Wir wollen dir diese notwendigen Mühen nicht verschweigen. Unserer Erfahrung nach scheitert so mancher Einstieg genau an dieser Hürde.

Unser Ansatz ist deshalb, durch **viele spannende Programmieraufgaben** das nötige Material zum Trainieren bereitzustellen. Du wirst in diesem Buch im Wesentlichen Aufgabenstellungen von uns bekommen, an denen du Aufgabe für Aufgabe alle relevanten Programmierkonzepte üben kannst. Damit dir dabei nicht die Laune vergeht, haben wir uns viel Mühe beim Zusammenstellen der Aufgaben gegeben. Es wird deutlich über die meisten „Lehrbuchaufgaben“ hinausgehen und sich, soweit möglich, erheblich näher an praktischen Anwendungsszenarien orientieren. Der klassische Lehrbuchstil handelt sich meist an Aufgabenstellungen aus der Mathematik entlang. Das ist wichtig, und daher haben auch wir das ab und an mit dir vor. Wir können aber auch verstehen, wenn derartige Aufgaben nicht jedem liegen, um etwas Neues zu lernen. Daher programmierst du eher Anwendungen, die etwas Nützliches tun oder etwas hübsch Anzuschauendes generieren. Als Appetitanreger haben wir im nachfolgenden Bild 1.1 schon mal drei Beispiele aus dem Buch für dich.

Diese drei Bilder zeigen exemplarisch, was du mit unserem Trainingsprogramm programmieren sollst und auch können wirst, wenn du fleißig am Ball bleibst. Es lohnt sich!

■ 1.3 Was muss ich mitbringen?

Nicht viel! Mit diesem Buch können wir es nicht leisten, dir die Basics beizubringen. Das musst du halt selbst tun oder du bekommst es in irgendeiner Form gezeigt. Wir erklären zu

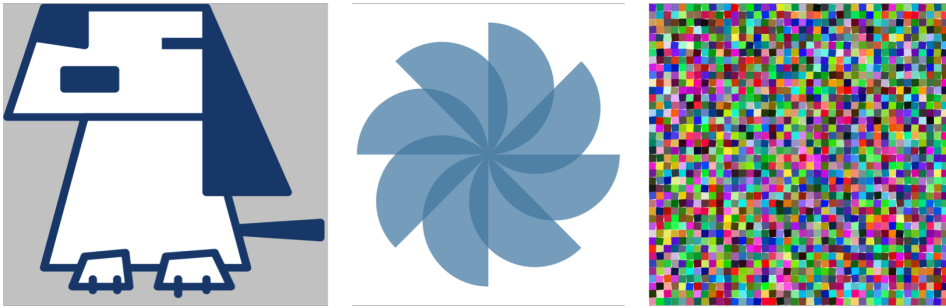


Bild 1.1 Drei Beispielbilder, die du programmieren wirst

Beginn eines jeden Kapitels nochmal kurz die im Fokus stehenden Übungsschwerpunkte. Das ist mehr eine Gedächtnisstütze und soll als Warm-up dienen, falls du es überhaupt brauchst. Wir gehen dabei nochmals kurz auf die wesentlichen Konzepte ein und erläutern Besonderheiten in den Programmierumgebungen, für die wir Beispiellösungen bereitstellen.

■ 1.4 Wie geht das vonstatten?

Wie schon gesagt, ist das hier ein Trainingsbuch fürs Programmieren. Wir stellen dir über **120 Übungsaufgaben mit Lösungsvorschlägen** zum Training bereit. Die grundlegende Struktur gleicht dabei derer gängiger Ressourcen zur Einführung in die Programmierung. Es geht mit dem Aufbau erster einfacher Programme los und wird durch das Hinzukommen von Programmierkonzepten wie Variablen, Datentypen, Operatoren, Ausdrücke, bedingte Anweisungen, Wiederholungsanweisungen, Funktionen, Arrays, Strings bis hin zur Objekt-orientierung stetig erweitert. Damit wir uns – ohne unnötiges Störfeuer und Ablenkung – auf das Kernthema des jeweiligen Kapitels konzentrieren können, bestehen die ersten Programme der Kapitel 2, 3 und 4 zunächst aus Anweisungen. Erst in den nachfolgenden Kapiteln 5 bis 9 kommen dann Strukturelemente für den Programmcode in Form von Funktionen sowie Klassen und Objekten hinzu. Wir werden dir Kapitel für Kapitel Trainingsaufgaben stellen, für die du dir ein passendes Programm überlegen und dieses dann in einer Programmiersprache vollständig angeben sollst.

Jede Trainingsaufgabe ist nach einem **festen Schema** aufgebaut (siehe Bild 1.2). Um jede Aufgabe eindeutig identifizieren zu können, haben wir diese mit einem eindeutigen Namen, einer eindeutigen Nummer und einem Piktogramm versehen. Das wird dir insbesondere dabei helfen, dich mit Freunden, Mitschülern, Kommilitonen, Kollegen oder der Community über die Aufgaben auszutauschen. Auch unsere Lösungsvorschläge im Anhang und online wirst du auf diese Weise spielend der jeweiligen Aufgabe zuordnen können. Die Identifizierungsnummer ist dem jeweiligen Buchkapitel zugeordnet. Das einfache Zurückspringen zur Aufgabenstellung im Buch ist damit auch gewährleistet.

Die Aufgaben haben wir unserem Dafürhalten nach in den Kategorien **Schwierigkeit**, **Kreativität** und **Zeitaufwand** bewertet und sortiert. Einfachere Aufgaben, die im Vergleich eher wenig Zeit und kreative Eigenleistung erfordern, findest du eher am Anfang eines jeden Kapitels. Du bist aber völlig frei in deiner Entscheidung, welcher Aufgabe du dich in welcher

2.2 Workshop 13	14 2. Einführung in die Programmierung
<p>2.2.2 Weihnachtsbaum</p> <p>Schwierigkeit: Zeitaufwand: Kreativität: </p> <p></p> <p>Themen</p> <p>Mit dieser Aufgabe wollen wir folgende Dinge trainieren:</p> <ul style="list-style-type: none"> • Struktur eines einfachen Programms • Aufbau und Abfolge von Programmieranweisungen • Ausgabe in der Konsole <p>Beschreibung</p> <p>Wir wollen jetzt ein erstes Muster in die Konsole schreiben. Dafür werden wir bestimmte Zeichen so oft hinter- und untereinander schreiben, bis sich daraus eine Form ergibt. Diese Form des "Malens" ist bei vielen Konsolenprogrammen üblich und wird auch heute noch verwendet.</p> <p>Aufgabenstellung</p> <p>Schreibe ein Programm, das das folgende Muster in der Konsole ausgibt:</p> <pre> * *** ***** ***** ***** ***** ***** ***** ***** ***** </pre> <p>Testfälle</p> <p>Wenn die Tanne wie angegeben in der Konsole ausgegeben wird, dann hast du alles richtig gemacht und diese Aufgaben erfolgreich bearbeitet. Gehezt den Fall, dass du noch weitere Programme diesen Typs erstellen willst, geben wir dir hier noch weitere Anregungen (kannst dir aber auch gerne selbst was überlegen!):</p> <p>Sanduhr: ***** Pizzastück: ***** Diamant: **</p> <pre> *** * * * * ** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * </pre>	<p>Für diese zusätzlichen Trainingseinheiten bieten wir die keine Lösungsvorschläge mehr an. Wir sind fest davon überzeugt, dass Du das selbst hinkommst und unsere Hilfe hierfür nicht mehr benötigst.</p> <p>Algorithmische Tipps</p> <p>Wenn Du stockst und nicht weiter weißt, dann versuch' mal folgendes:</p> <ul style="list-style-type: none"> • Schau' dir die Aufgabe 2.2.1 doch nochmal an und überlege Dir, wie die Ausgabe für jede Zeile von oben nach unten aussehen muss. • In Processing für Java gibt es zwei Befehle, mit denen du Text in die Konsole schreiben kannst. Der eine fügt eine neue Zeile hinzu, der andere hingegen nicht. • Das Sternchenzeichen und Leerzeichen führen zum Ziel!

Bild 1.2 Exemplarische Darstellung des Aufgaben-Templates

Reihenfolge widmen möchtest. Du musst auch längst nicht alle Aufgaben durchhackern. Wenn der Groschen in Bezug auf die in einem Kapitel fokussierten Programmierkonzepte gefallen ist, kann es ans nächste Kapitel gehen.

Wir haben schon eingangs gesagt, dass wir mit dir und den Aufgaben eines jeden Kapitels bestimmte Programmierkonzepte trainieren möchten. Auf welche **Themen** sich eine jeweilige Aufgabe dabei im Besonderen fokussiert, geben wir dir kurz stichwortartig an. Dies soll ein zusätzliches Kriterium sein, wonach du entscheiden kannst, ob du eine bestimmte Aufgabe bearbeiten möchtest.

Worum es sich in einer Aufgabe dreht, erläutert eine kurze **Beschreibung**. Hier wird der Kontext gesetzt und ggf. auch ein Praxisbezug hergestellt, damit du weißt, wo sich in der Praxis etwas Derartiges finden lässt. Was du dann tatsächlich tun musst, geben wir dir im anschließenden Abschnitt mit dem passenden Namen **Aufgabenstellung** an. Du solltest beide Abschnitte aufmerksam und eventuell mehrmals lesen, um sicherzustellen, dass du deinen Auftrag richtig verstanden hast.

Zur Überprüfung, ob dein Programm tatsächlich funktioniert, geben wir dir **Testfälle** mit entsprechenden Testdaten an die Hand. Zudem haben wir zu jeder Aufgabe eine ausführlich kommentierte Lösung in den Programmiersprachen Java und Python beigelegt. Bevor du dir die aber anguckst, solltest du wirklich lange Zeit selbst an der Erarbeitung einer Lösung werkeln.

Wenn du völlig auf dem Schlauch stehst und gar keinen Zugang findest, sind am Ende einer jeden Aufgabe **algorithmische Tipps** aufgelistet, die dich einem möglichen Lösungsansatz näherbringen sollen. Bitte nutze diese Tipps und versuche unbedingt, selbst eine Lösung

herzustellen, bevor du dir unsere Lösungsvorschläge im Anhang des Buchs anschaust. Nur auf diese Weise kommst du genügend ins Schwitzen, um nachher wirklich behaupten zu können, das Programmieren auch selbstständig zu beherrschen.

■ 1.5 Was muss ich sonst noch wissen?

Damit du möglichst effektiv und fokussiert trainieren kannst, haben wir die Trainingsumgebung für dich von unnötigem Ballast entschlackt. Du sollst nicht schon bei der Installation, Konfiguration und Verwendung der Programmierumgebung die Lust am Programmieren bzw. die Sicht auf das Wesentliche verlieren. Wir stützen uns daher auf ein einziges Werkzeug, mit dem du in **Java und Python** das Programmieren trainieren kannst. Es handelt sich dabei um das frei und kostenlos verfügbare Tool mit dem Namen **Processing**, das du unter der Web-Adresse <https://processing.org/> abrufen kannst. Hier findest du auch viele weitere Informationen und Dokumentationen rund um Processing. Im Anhang C haben wir Installationsanleitungen für die gängigen Betriebssysteme Windows, macOS und Linux beigelegt.

Dass wir auf Processing abstellen, soll aber nicht heißen, dass du mit dem Erlernten in der Praxis nicht viel anfangen kannst. Ganz im Gegenteil! Die Programmiersprachen Java und Python gehören zu den am weitesten verbreiteten Sprachen, und schließlich kommt es im Wesentlichen auf die Programmierkonzepte an. Wenn du diese intensiv trainiert und dadurch verinnerlicht hast, dann bist du bereit, alle möglichen Aufgabenstellungen programmatisch zu lösen. Dann haben wir unser gemeinsames Ziel erreicht. Die Verwendung professionellerer Programmierumgebungen wie z. B. Eclipse, IntelliJ oder PyCharm ist dann ein Klacks. Darüber müssen wir dann nicht mehr viele Worte verlieren. Wenn Du so gar keine Lust auf Processing haben solltest und unsere Aufgaben lieber mit anderen Programmierumgebungen für Java oder Python bearbeiten möchtest, dann findest Du im Anhang D Anleitungen für den Aus- und Umstieg.

Die Quelltexte zum Buch – unsere Lösungsvorschläge – haben wir auf dem Onlinedienst **GitHub** für dich bereitgelegt. Du findest sie unter der Adresse <https://github.com/protrain>. Wie du damit umgehst, erklären wir für alle gängigen Desktop-Betriebssysteme im Anhang A.1 (für Java) bzw. im Anhang B.1 (für Python).

Mit dem Kauf des Buches soll aber noch nicht Schluss sein. Wir würden uns sehr freuen, von dir zu hören. Über GitHub kannst du uns z. B. auf Fehler im Buch oder in den Lösungen aufmerksam machen. Wir tragen das dann in die Errata-Liste ein bzw. korrigieren die Programme. Außerdem kannst du uns auch deine Lösung(en) bereitstellen. Wenn diese einen eigenen Lösungsweg zeigen, nehmen wir sie mit in das Repositorium auf. Selbiges gilt für Lösungen in anderen Programmiersprachen. Achte aber bitte hierbei darauf, dass es für die Sprache eine ähnliche einfache und umfangreiche Programmierumgebung gibt, wie es Processing für Java und Python ist. Wenn du sonstige Anregungen zur Verbesserung hast oder Ideen für weitere Aufgaben beisteuern möchtest, freuen wir uns von dir zu hören.

Hoffentlich konnten wir dein Interesse wecken und dir unseren Ansatz zum Programmieren lernen schmackhaft machen. Jedenfalls würden wir uns sehr freuen, gemeinsam mit dir das Programmieren zu trainieren.



2

Einführung in die Programmierung

■ 2.1 Warmup

Dein Training beginnt in diesem Kapitel mit ersten einfachen Programmen. Dazu musst du wissen, wie der grundlegende Aufbau eines Programms sowie der Aufbau der Anweisungen in einer bestimmten Programmiersprache ist. Letzteres gehört zur sogenannten **Syntax** einer Programmiersprache. So wie z. B. die Syntax einer natürlichen Sprache Prinzipien und Regeln des Wort- und Satzbaus festlegt, so legt die Syntax einer Programmiersprache das Vokabular und den Aufbau von Anweisungen fest.

Für die allerersten Programme, die du entwickeln sollst, genügt zunächst die allereinfachste Struktur überhaupt. Hierbei werden Programme als eine lineare Abfolge von Anweisungen angegeben. Anweisungen verfügen immer über einen Namen und eine Liste von Parametern, die die Anweisung verarbeiten soll. Um den Anweisungsnamen von der Parameterliste unterscheiden zu können, werden die Parameter häufig eingeklammert und dem Anweisungsnamen nachgestellt.

```
nameAnweisung(parameter);
```

Verfügt die Parameterliste über mehrere Einträge, so werden diese mit Komma (,) voneinander getrennt.

```
nameAnweisung(parameter1, parameter2);
```

Parameterlose Anweisungen sind durch ein leeres Klammernpaar gekennzeichnet.

```
nameAnweisung();
```

Um mehrere Anweisungen voneinander unterscheiden zu können, wird dafür ein Trennzeichen in der Syntax einer Programmiersprache festgelegt. In Java ist das das Semikolon (;). Das folgende Beispiel zeigt ein abstraktes Programm, das sich aus sieben Anweisungen zusammensetzt, die in der angegebenen Reihenfolge ausgeführt werden. Die lineare Programmabfolge führt die programmierten Anweisungen zeilenweise von links nach rechts beginnend mit der obersten Zeile aus.

```
Anweisung1(); Anweisung2(); Anweisung3(); Anweisung4(); Anweisung5();  
Anweisung6(); Anweisung7();
```

Durch diese Syntaxregel können die einzelnen Anweisungen separiert werden, unabhängig davon, wie du diese in die Quelltextdatei schreibst. Zur besseren Lesbarkeit empfehlen wir dir aber, dich auf eine Anweisung pro Zeile zu beschränken und die Anweisungen untereinander zu schreiben.

```
Anweisung1();
Anweisung2();
Anweisung3();
Anweisung4();
Anweisung5();
Anweisung6();
Anweisung7();
```

Die Programmiersprache Python legt in ihrer Syntax als Trennzeichen von Anweisungen den Zeilenumbruch fest. Ein Zeilenumbruch kann je nach Betriebssystem aus einem oder zwei Zeichen bestehen ('\n', '\r' oder '\r\n').

In der Programmierliteratur hat sich das "Hello World!"-Programm als einführendes Beispiel zur Darstellung der grundlegenden Syntax eines einfachen Programms in einer bestimmten Programmiersprache etabliert. Das Hello-World-Programm gibt in der Konsole einen einfachen Text aus, nämlich Hello World!. Wir wollen es zur Konkretisierung der einführenden Erläuterungen verwenden.

Java:

```
print("Hallo_Welt!");
```

Python:

```
print("Hallo_Welt!")
```

Die `print()`-Anweisung bekommt einen Parameter übergeben. Dieser enthält den Text, den die Anweisung in der Konsole ausgeben soll. Um den Text eingrenzen zu können, wird dieser von doppelten Anführungszeichen (") eingerahmt.

Die Aufgaben dieses Kapitels drehen sich um derartige Programme. Deine Aufgabe wird es sein, die zur Lösung der Aufgabenstellung benötigten Anweisungen zu identifizieren und diese dann in einer geeigneten Abfolge zu platzieren. Welche Anweisungen eine Programmiersprache im Standardumfang bereitstellt, sind in der Referenzdokumentation aufgeführt. Die Referenz der von Processing bereitgestellten Anweisungen kann im Internet eingesehen werden:

- <https://processing.org/reference/> (Java)
- <http://py.processing.org/reference/> (Python)

Referenzen sind sehr umfangreich. Dies gilt auch für die von Processing. Es kann daher etwas dauern, bist du dich darin zurechtfindest. Für die in diesem Kapitel bereitgestellten Trainingsaufgaben sind insbesondere Funktionen zur Ausgabe von Texten in der Konsole und Funktionen zur Ausgabe elementarer geometrischer Formen im grafischen Ausgabefenster wichtig. Um dir das Auffinden dieser Anweisungen zu erleichtern, führen wir dir in der nachfolgenden Auflistung die relevanten auf.

- Konsolenausgabe
 - https://processing.org/reference/print_.html (Java)
 - <http://py.processing.org/reference/print.html> (Python)
- Linie
 - https://processing.org/reference/line_.html (Java)
 - <http://py.processing.org/reference/line.html> (Python)

- Dreieck
 - https://processing.org/reference/triangle_.html (Java)
 - <http://py.processing.org/reference/triangle.html> (Python)
- Rechteck
 - https://processing.org/reference/rect_.html (Java)
 - <http://py.processing.org/reference/rect.html> (Python)
- Viereck
 - https://processing.org/reference/quad_.html (Java)
 - <http://py.processing.org/reference/quad.html> (Python)
- Ellipse
 - https://processing.org/reference/ellipse_.html (Java)
 - <http://py.processing.org/reference/ellipse.html> (Python)
- Kreisausschnitt
 - https://processing.org/reference/arc_.html (Java)
 - <http://py.processing.org/reference/arc.html> (Python)

Um sich mit der Funktionsweise der Anweisungen vertraut zu machen, empfehlen wir dir, die Beschreibung in der Referenz aufmerksam zu lesen. Dies ist eine wichtige Grundfertigkeit, die zum Programmieren dazugehört.

Verwenden werden wir in diesem Buch die Entwicklungsumgebung Processing. Hiermit können wir Programme sowohl in Java als auch in Python schreiben. Processing bietet nicht nur den Vorteil der einfachen Installation auf nahezu allen Betriebssystemen. Wir können damit auch sehr einfach (grafische) Programme auf Basis von Anweisungen schreiben. Aber auch höherwertige Konzepte, wie wir sie in den späteren Kapiteln umsetzen werden, sind in Processing möglich. Perfekte Voraussetzungen also zum Trainieren deiner Programmiertechniken mit diesem Buch.

Alle Installationsschritte von Processing findest du in Anhang C.1. Wie du an die digitalen Quelltexte unserer Lösungsvorschläge zu einzelnen Aufgaben kommst und wie du sie in Processing öffnest, steht im Anhang A.1.1 für Java und im Anhang B.1.1 für Python.

Dateien mit Quelltext können wir in Processing mit Klick auf *Datei* → *Öffnen* ... laden. In Bild 2.1 haben wir zum Beispiel eine solche Datei geöffnet. Dort können wir gut die grafische Benutzeroberfläche von Processing erkennen:

- Mit dem Start- und Stopp-Button (1) kannst du deinen Java- bzw. Python-Code ausführen.
- Um vom Java- auf den Python-Modus zu wechseln, kannst du den Modus-Auswahltreiter (2) verwenden. Wie du den Python-Modus in Processing installierst, steht in Anhang C.5. Links neben diesem Button ist der integrierte Debugger, den du zur Analyse von Java-Code verwenden kannst. Mehr dazu findest du in Anhang A.1.3.
- In der Mitte der Benutzeroberfläche (3) steht der eigentliche Quelltext. In diesen Bereich kannst du deinen Java- bzw. Python-Code hineinschreiben.
- Entsprechende Ausgaben in der Konsole findest du im darunterliegenden Bereich (4). Hier werden auch auftretende Fehler im Code angezeigt, sofern es welche gibt.

Nach der Einrichtung von Processing und dem Lesen der Einführung solltest du für dieses Kapitel ausgerüstet sein. In dem Sinne: Viel Spaß bei den ersten Aufgaben!