

Inhaltsverzeichnis

	Über den Autor	15
	Vorwort	17
Teil 1	Swift: Der neue Stern unter den Programmiersprachen	
1	Wie Sie Swift lernen	21
1.1	Grafische Oberflächen auf Apple-Systemen	21
1.2	Die Geschichte von Swift	22
1.3	Was Sie hier programmieren werden	23
1.4	Wenn Sie bereits Objective-C kennen	23
1.5	Was Sie mitbringen müssen	24
1.6	Wie Sie mit diesem Buch Erfolg haben	24
Teil 2	Arbeiten mit Xcode: Die ersten Schritte	
2	Ihre erste App auf dem iPhone!	29
2.1	Warum Sie überhaupt Programmiersprachen brauchen	29
2.2	Xcode von Apple	30
2.3	Die richtigen Anlagen für Ihr erstes Projekt	32
2.4	Hallo Welt	39
2.5	Der iPhone-Simulator	42
2.6	Was Xcode sonst noch alles kann	43
2.7	Ausgaben: Sie können auch anders	48
2.8	Aufgaben	50
3	Mit Zahlen und Buchstaben: Mit Ihrem Mac können Sie rechnen	51
3.1	Playgrounds: Die Spielwiese verwenden	52
3.2	Variablen: Es kommt auf die Werte an	55
3.3	Datentypen: Von Zeichen, Texten und Zahlen	59
3.3.1	Datentypen deklarieren und initialisieren	61
3.3.2	Strings im Programm	62
3.3.3	Fließkommazahlen: Punkt, Punkt, Komma, Strich	62

3.4	Bool – Der (Daten-)Typ der Wahrheit	64
3.4.1	Zahlen vergleichen – alles ist relativ	68
3.4.2	Operationen mit Bool – Rechnen mit Wahrheiten	69
3.5	Lesbaren Programmtext schreiben	71
3.5.1	Gute Kommentare	71
3.5.2	Formatieren von Programmtext und Namensgebung	74
3.6	Fortgeschrittenes	77
3.6.1	Unterschiede zwischen Zahlentypen	77
3.7	Aufgaben	78

Teil 3 Programme in Swift

4	Die Kommandozeile, Variablen und Gültigkeit	83
4.1	Kommandozeilenprogramme: Eigenständige Programme unter MacOS X	84
4.1.1	Das Projekt in Xcode erstellen	84
4.1.2	Programme unabhängig von Xcode starten	87
4.1.3	Ausgabe von Daten	89
4.1.4	Eingabe von Daten	90
4.1.5	Praxisbeispiel: Der ganz flexible Zinsrechner – das zahl ich Ihnen zurück	93
4.1.6	Grammatisch korrekte Sätze	97
4.2	Funktionen: Definieren und richtig benutzen	99
4.2.1	Funktionen mit Rückgabewerten	99
4.2.2	Funktionen mit Parametern	101
4.3	Gültigkeit	102
4.4	Fortgeschrittenes	104
4.4.1	Bits und Bytes	104
4.4.2	Speicher: Wie Ihr Computer sich Dinge merkt	105
4.4.3	Polymorphe Funktionen	107
4.4.4	Skripte von der Kommandozeile	108
4.5	Aufgaben	108
5	Praxisbeispiele: Von Trinkspielen, Sparplänen und korrekter Grammatik	109
5.1	Praxisbeispiel: Fizz Buzz	110
5.1.1	Das Grundgerüst	110
5.1.2	Teilbarkeit und der ganze Rest	111
5.1.3	Die fertige Funktion	112
5.2	Sparpläne mit Funktionen: Was lohnt sich und was nicht?	114

5.3	Alternativen mit der switch-Anweisung	118
5.3.1	Die Grundform der switch-Anweisung	118
5.3.2	Ich stelle hier die Bedingungen: die where-Anweisung	120
5.4	Was Funktionsparameter alles können	121
5.4.1	Die richtigen Namen von Funktionsparametern	121
5.4.2	Funktionen mit mehreren Rückgabewerten	123
5.4.3	Parameter als Variablen	126
5.4.4	Vorgegebene Standardwerte	127
5.5	Fortgeschrittenes	129
5.5.1	Wenn die Werte mal nicht so wichtig sind	129
5.6	Aufgaben	130
6	Ihr erstes Spiel programmieren: Das Streichholzspiel	133
6.1	Züge im Streichholzspiel	134
6.1.1	Zufallszahlen erzeugen	134
6.2	Züge des menschlichen Spielers	138
6.2.1	Schleifen mit der while-Anweisung	138
6.3	Das fertige Spiel	141
6.3.1	Ich will raus! – Schleifen mit break verlassen	141
6.4	Refaktorisieren – Verschönern Ihrer Programme	144
6.5	Usability – Der Spaßfaktor	147
6.6	Fortgeschrittenes	150
6.6.1	Kontrolle – Ihre Schleife, bitte	150
6.6.2	Schleifen und Sprünge für Fortgeschrittene	152
6.6.3	Auf Nummer sicher gehen mit der assert-Anweisung	154
6.6.4	Mehrere Anweisungen pro Zeile	154
6.7	Aufgaben	155

Teil 4 Objekte und Bibliotheken mit Swift

7	Objekte – Was sind sie und wie können sie Ihnen helfen, Arbeit zu sparen?	159
7.1	Objekte in Swift	159
7.1.1	Definition von Objekten	160
7.1.2	Verwendung von Objekten	161
7.1.3	Objektinstanzen im Speicher	162
7.2	Praxisbeispiel: Das Unternehmerspiel	166
7.2.1	Anlegen des Projekts und Ihres ersten Objekts	166
7.2.2	Das Unternehmer-Objekt	168
7.2.3	Die Methoden des Unternehmer-Objekts	170
7.2.4	Das Hauptprogramm	172

7.3	Aufzählungen als Datentypen	176
7.4	Fortgeschrittenes	179
7.4.1	Privatsphäre für Klassen	179
7.4.2	Eigenschaften von Klassen unter Beobachtung	180
7.4.3	Berechnete Eigenschaften	181
7.5	Aufgaben	183
8	Wie Sie das System mit Cocoa-Objekten voll ausreizen	185
8.1	So holen Sie Hilfe	186
8.2	NSString: Dieses Wort ist mächtiger als das Schwert	189
8.2.1	Zahlen, Formate und Ersetzungen	190
8.2.2	Vergleiche von Strings	191
8.2.3	Suchen und Ersetzen	194
8.3	Praxisbeispiel: Schon wieder ein neues Passwort	194
8.3.1	Ein einzelnes Zeichen setzen	195
8.3.2	Lottospielen mit Ziffern	196
8.3.3	Jetzt aber richtig: Großbuchstaben, Kleinbuchstaben und alles zusammen	198
8.4	Fortgeschrittenes	201
8.4.1	Formatierte Ausgabe	201
8.4.2	Wie Strings gespeichert werden	202
8.4.3	Unicode	203
8.5	Aufgaben	204
9	Der Lebenszyklus von Objekten	207
9.1	Objekte erzeugen und vernichten	208
9.1.1	Erzeugen eines Objekts	208
9.1.2	Deinitialisieren eines Objekts	210
9.2	Objekte als Eigenschaften	212
9.2.1	Optionale Datentypen	215
9.2.2	Starke Typen: Schwache Zeiger	217
9.2.3	Objekte ohne Besitzer	224
9.2.4	Zusammenfassung	226
9.3	Kopieren von Objekten: Der Weg von Swift	227
9.4	Fortgeschrittenes	230
9.4.1	Operatorüberladung	230
9.5	Aufgaben	231
10	Verladen von Objekten – Alles über Datencontainer	233
10.1	Arrays – sortiert nach Zahlen	234
10.1.1	Arrays erzeugen	234
10.1.2	Arrays verändern	235
10.2	FizzBuzz mit Arrays	237

10.3	Dictionaries – Informationen sortiert abspeichern	239
10.3.1	Dictionaries erzeugen und ändern	240
10.3.2	Dictionaries benutzen	241
10.4	Beliebige Werte	242
10.5	Daten speichern und laden	244
10.5.1	Wo geht’s hier zum Verzeichnis?	244
10.5.2	Daten speichern und wieder laden	245
10.5.3	Ich kann mir nicht alles merken!	246
10.6	Fortgeschrittenes	250
10.6.1	Funktionale Programmierung	250
10.6.2	Generische Klassen und Funktionen	253
10.7	Aufgaben	254
11	Wie Sie sich mit Vererbung bei Objekten richtig Zeit und Mühe sparen	257
11.1	Wie geht Vererbung?	257
11.1.1	Vererbung: Funktionsreichtum ohne Mehrarbeit	259
11.1.2	Den Arbeiter in das Unternehmensspiel aufnehmen	262
11.1.3	Für die Einstellung vorbereiten: Den Angestellten definieren	264
11.1.4	Die Arbeit mit dem neuen Mitarbeiter	265
11.2	Polymorphismus ist keine Krankheit, sondern gut für Sie	267
11.2.1	Das Spiel mit Praktikanten	268
11.3	Protokolle: Gemeinsamkeiten über Klassengrenzen hinweg	270
11.4	Fortgeschrittenes	273
11.4.1	Das letzte Wort, Verbieten von Vererbung	274
11.4.2	Erweiterungen von Klassen	274
11.4.3	Benutzung von NSObject	275
11.4.4	Klassenmethoden	275
11.4.5	SOLID – Solide Programme bauen	277
11.5	Aufgaben	277

Teil 5 Grafische Oberflächen für das iPhone

12	Jetzt schreiben Sie grafische Oberflächen auf dem iPhone	281
12.1	Button, übernehmen Sie	281
12.1.1	Den Button und den Slider einfügen	282
12.1.2	Der Simulator kriegt den Drehwurm	283
12.1.3	Ordnung ist das halbe Leben: Anordnen und Ausrichten	285
12.2	Drehungen erlauben: Auf den Kopf stellen bringt nichts	290
12.3	Die grafische Oberfläche in der App verwenden	292
12.3.1	Der Button bekommt Arbeit	293

12.3.2	Es geht auch anders herum: Das UI als Eigenschaft	297
12.3.3	Beispiel: Die Beschriftung ändern	300
12.3.4	Beispiel: Den Button deaktivieren	300
12.3.5	Schiebung mit dem Regler: Arbeiten mit Slidern	301
12.3.6	Den Slider näher kennenlernen: Seine Grenzen kennen und Voreinstellungen	303
12.4	Ausrichtungsregeln löschen und ändern	303
12.5	Fortgeschrittenes	306
12.5.1	Apps für die ganze Welt	306
12.6	Aufgaben	309
13	Tabellen und Programmfluss auf dem iPhone	311
13.1	Tabellen auf dem iPhone	311
13.2	Praxisbeispiel: FizzBuzz mit Tabellen	313
13.2.1	Die Vorlage für Tabellen	313
13.2.2	Das Einrichten der Datenquelle	316
13.2.3	Super: Aufrufen der Basisklasse	320
13.2.4	Ganze Geschichten erzählen: Den zweiten Bildschirm ver- wenden	320
13.2.5	Das Ergebnis: Wenig Arbeit für Sie	329
13.3	MVC – Das Geheimrezept, mit dem Ihre Programme richtig elegant werden	329
13.4	Aufgaben	333
14	Nun sind Sie dran: Zwei grafische Apps auf dem iPhone	335
14.1	Praxisbeispiel: Wie Sie Ihren eigenen Webbrowser schreiben	336
14.1.1	Präsentation	337
14.1.2	Die Steuerung	340
14.1.3	Besonderheiten unter iOS 9	343
14.1.4	Der fertige Netzbrowser	345
14.2	Passwörter	348
14.2.1	Modell der Passwörter	348
14.2.2	Präsentation der Passwörter	352
14.2.3	Steuerung der Passwörter	354
14.2.4	Wie Sie Ihre App richtig schön machen: Icons und Grafiken .	357
14.3	Fortgeschrittenes	362
14.3.1	Arbeiten im Hintergrund	362
14.3.2	Was Textfelder alles können	364
14.4	Aufgaben	364

15	Eine umfangreiche App: Das Streichholzspiel auf dem iPhone	367
15.1	Das Datenmodell	367
15.1.1	Das Modell für den Haufen	368
15.1.2	Die bisherige Strategie	370
15.2	Die grafische Oberfläche	371
15.2.1	Der Spielbildschirm	372
15.2.2	Der Konfigurationsbildschirm	374
15.3	Die Steuerung	376
15.3.1	Die erste lauffähige Version	376
15.3.2	Gehe zurück auf Los, ziehe Hölzer ein	381
15.4	Gewinnstrategien für Ihren Computer	385
15.4.1	Strategien mit Polymorphismus	385
15.4.2	Eine wirklich gute Strategie	386
15.4.3	Strategien auswählen	389
15.5	Der Konfigurationsbildschirm	392
15.6	Fortgeschrittenes	403
15.6.1	Arbeiter und Vermittler	403
15.6.2	Entwurfsmuster	404
15.7	Aufgaben	404
16	Die eigene Sicht der Dinge vermitteln: Eigene grafische Ausgaben und Animationen	407
16.1	Ich zeige Sie an – unter iOS	407
16.1.1	Größe und Position	408
16.1.2	Dynamische Größenanpassung	410
16.2	Der grafische Streichholzhaufen	411
16.2.1	Streichhölzer Variante 1	411
16.2.2	Streichhölzer Variante 2 – Sie können auch anders	415
16.3	Einen Haufen machen	420
16.4	Grafische Animationen	425
16.4.1	Was können Sie grafisch animieren?	425
16.4.2	Wie Sie Animation durchführen	426
16.4.3	Die Animation im Streichholzspiel	427
16.5	Lang lebe die Streichholz-App	430
16.6	Aufgaben	435

Teil 6 Grafische Oberflächen für iPad, die Apple Watch und MacOS X

17	Für das iPad entwickeln und mit echten Geräten arbeiten	439
17.1	Was macht das iPad besonders?	440

17.2	Passwörter auf dem iPad	441
17.2.1	Vom iPhone zum iPad	441
17.2.2	iPhone und iPad – eine App für alle Situationen	442
17.2.3	Eigene Wege gehen: Die andere Oberfläche für das iPad	445
17.2.4	Unterschiedliche Programme für das iPhone und iPad	447
17.3	Auf echten Geräten entwickeln	450
17.4	Fortgeschrittenes	455
17.4.1	Die Empfehlungen von Apple	455
18	Grafische Anwendungen unter MacOS X und auf der Apple Watch . .	457
18.1	Besonderheiten von MacOS X	458
18.2	Praxisbeispiel: Das Passwort-Programm mit grafischer Oberfläche unter MacOS X	459
18.2.1	Das Datenmodell	460
18.2.2	Die Präsentation	460
18.2.3	Die Steuerung	465
18.3	Besonderheiten der Apple Watch	467
18.4	Passwörter auf der Apple Watch	467
18.4.1	Das Datenmodell	469
18.4.2	Die Präsentation	469
18.4.3	Die Steuerung	470
18.5	Fortgeschrittenes	472
18.5.1	Empfehlungen von Apple für MacOS X und die Apple Watch	472
18.6	Aufgaben	472

Teil 7 Geld verdienen als Software-Ingenieur

19	Was Sie häufig brauchen: Entwanzen von Programmen	477
19.1	Typische Fallstricke – Arten von Fehlern	477
19.1.1	Syntaktische und semantische Fehler	478
19.1.2	Usability-Probleme	478
19.1.3	Falsche Ergebnisse von Programmen	479
19.1.4	Nicht reproduzierbare Fehler	479
19.1.5	Umgang mit Fehlern	480
19.2	Verwenden des Debuggers	482
19.2.1	Praxisbeispiel: Ein fehlerhaftes FizzBuzz-Programm	483
19.3	Anforderungen	491
19.4	Testen, Testen und nochmals Testen	492
19.4.1	Was automatisierte Tests tun	492
19.4.2	Praxisbeispiel: Das HolzHaufen-Modell des Streichholzspiels	493
19.4.3	Wie Sie Tests noch einsetzen können	497

19.5	Fortgeschrittenes	498
	19.5.1 Ausnahmen und Fehlerbehandlung	498
	19.5.2 Versionsmanagement	501
19.6	Aufgaben	502
20	Der Weg zum Software-Ingenieur	503
20.1	Vom Programmierer zum Software-Ingenieur	503
20.2	Arbeiten als Software-Ingenieur	504
20.3	Vorschläge formulieren und Lösungen unterbreiten	505
20.4	Wie lange dauert das?	507
20.5	Der wöchentliche Statusbericht	508
20.6	Kontakte knüpfen und gute Fragen stellen	509
A	Lösung der Aufgaben	513
	Stichwortverzeichnis	537