

Inhalt

Vorwort	XXIII
Teil I: Swift	1
1 Die Programmiersprache Swift	3
1.1 Die Geschichte von Swift	4
1.2 Swift-Updates	5
1.3 Voraussetzungen für die Nutzung von Swift	6
1.4 Installation von Swift	7
1.4.1 macOS	7
1.4.2 Linux	10
1.5 Playgrounds	13
1.5.1 Erstellen eines Playgrounds	14
1.5.2 Aufbau eines Playgrounds	16
1.5.3 Pages, Sources und Resources	20
1.5.4 Playground-Formatierungen	22
1.5.5 Swift Playgrounds-App für das iPad	31
2 Grundlagen der Programmierung	35
2.1 Grundlegendes	35
2.1.1 Swift Standard Library	35
2.1.2 print	37
2.1.3 Befehle und Semikolons	38
2.1.4 Operatoren	39
2.2 Variablen und Konstanten	40
2.2.1 Erstellen von Variablen und Konstanten	40
2.2.2 Variablen und Konstanten in der Konsole ausgeben	41
2.2.3 Type Annotation und Type Inference	42
2.2.4 Gleichzeitiges Erstellen und Deklarieren mehrerer Variablen und Konstanten	44
2.2.5 Namensrichtlinien	44
2.3 Kommentare	45

3	Schleifen und Abfragen	47
3.1	Schleifen	47
3.1.1	for-in	47
3.1.2	while	49
3.1.3	repeat-while	50
3.2	Abfragen	51
3.2.1	if	51
3.2.2	switch	55
3.2.3	guard	59
3.3	Control Transfer Statements	60
3.3.1	Anstoßen eines neuen Schleifendurchlaufs mit continue	61
3.3.2	Verlassen der kompletten Schleife mit break	61
3.3.3	Labeled Statements	62
4	Typen in Swift	65
4.1	Integer	66
4.2	Fließkommazahlen	68
4.3	Bool	69
4.4	String	69
4.4.1	Erstellen eines Strings	69
4.4.2	Zusammenfügen von Strings	70
4.4.3	Character auslesen	71
4.4.4	Character mittels Index auslesen	72
4.4.5	Character entfernen und hinzufügen	73
4.4.6	Anzahl der Character zählen	75
4.4.7	Präfix und Suffix prüfen	75
4.4.8	String Interpolation	75
4.5	Array	76
4.5.1	Erstellen eines Arrays	77
4.5.2	Zusammenfügen von Arrays	78
4.5.3	Inhalte eines Arrays leeren	78
4.5.4	Prüfen, ob ein Array leer ist	79
4.5.5	Anzahl der Elemente eines Arrays zählen	79
4.5.6	Zugriff auf die Elemente eines Arrays	80
4.5.7	Neue Elemente zu einem Array hinzufügen	80
4.5.8	Bestehende Elemente aus einem Array entfernen	81
4.5.9	Bestehende Elemente eines Arrays ersetzen	82
4.5.10	Alle Elemente eines Arrays auslesen und durchlaufen	83
4.6	Set	84
4.6.1	Erstellen eines Sets	84
4.6.2	Inhalte eines bestehenden Sets leeren	85
4.6.3	Prüfen, ob ein Set leer ist	86
4.6.4	Anzahl der Elemente eines Sets zählen	86
4.6.5	Element zu einem Set hinzufügen	86

4.6.6	Element aus einem Set entfernen	87
4.6.7	Prüfen, ob ein bestimmtes Element in einem Set vorhanden ist	87
4.6.8	Alle Elemente eines Sets auslesen und durchlaufen	87
4.6.9	Sets miteinander vergleichen	88
4.6.10	Neue Sets aus bestehenden Sets erstellen	91
4.7	Dictionary	92
4.7.1	Erstellen eines Dictionaries	93
4.7.2	Prüfen, ob ein Dictionary leer ist	94
4.7.3	Anzahl der Schlüssel-Wert-Paare eines Dictionaries zählen	94
4.7.4	Wert zu einem Schlüssel eines Dictionaries auslesen	94
4.7.5	Neues Schlüssel-Wert-Paar zu Dictionary hinzufügen	95
4.7.6	Bestehendes Schlüssel-Wert-Paar aus Dictionary entfernen	96
4.7.7	Bestehendes Schlüssel-Wert-Paar aus Dictionary verändern	96
4.7.8	Alle Schlüssel-Wert-Paare eines Dictionaries auslesen und durchlaufen	97
4.8	Tuple	98
4.8.1	Zugriff auf die einzelnen Elemente eines Tuples	99
4.8.2	Tuple und switch	100
4.9	Optional	103
4.9.1	Deklaration eines Optionals	104
4.9.2	Zugriff auf den Wert eines Optionals	104
4.9.3	Optional Binding	106
4.9.4	Implicitly Unwrapped Optional	108
4.9.5	Optional Chaining	109
4.9.6	Optional Chaining über mehrere Eigenschaften und Funktionen ...	113
4.10	Any und AnyObject	117
4.11	Type Alias	118
4.12	Value Type versus Reference Type	118
4.12.1	Reference Types auf Gleichheit prüfen	120
5	Funktionen	123
5.1	Funktionen mit Parametern	124
5.1.1	Argument Labels und Parameter Names	125
5.1.2	Default Value für Parameter	128
5.1.3	Variadic Parameter	129
5.1.4	In-Out-Parameter	130
5.2	Funktionen mit Rückgabewert	131
5.3	Function Types	133
5.3.1	Funktionen als Variablen und Konstanten	134
5.4	Verschachtelte Funktionen	136
5.5	Closures	136
5.5.1	Closures als Parameter von Funktionen	138
5.5.2	Trailing Closures	141
5.5.3	Autoclosures	142

6	Enumerations, Structures und Classes	145
6.1	Enumerations	145
6.1.1	Enumerations und switch	148
6.1.2	Associated Values	149
6.1.3	Raw Values	151
6.2	Structures	154
6.2.1	Erstellen von Structures und Instanzen	154
6.2.2	Eigenschaften und Funktionen	155
6.3	Classes	160
6.3.1	Erstellen von Klassen und Instanzen	161
6.3.2	Eigenschaften und Funktionen	161
6.4	Enumeration vs. Structure vs. Class	163
6.4.1	Gemeinsamkeiten und Unterschiede	163
6.4.2	Wann nimmt man was?	164
6.5	self	166
7	Eigenschaften und Funktionen von Typen	169
7.1	Properties	169
7.1.1	Stored Property	170
7.1.2	Lazy Stored Property	172
7.1.3	Computed Property	176
7.1.4	Read-Only Computed Property	178
7.1.5	Property Observer	180
7.1.6	Type Property	183
7.2	Globale und lokale Variablen	185
7.3	Methoden	188
7.3.1	Instance Methods	189
7.3.2	Type Methods	191
7.4	Subscripts	192
8	Initialisierung	197
8.1	Aufgabe der Initialisierung	198
8.2	Erstellen eigener Initializer	199
8.3	Initializer Delegation	204
8.3.1	Initializer Delegation bei Value Types	205
8.3.2	Initializer Delegation bei Reference Types	206
8.4	Failable Initializer	208
8.5	Required Initializer	211
8.6	Deinitialisierung	212
9	Vererbung	215
9.1	Überschreiben von Eigenschaften und Funktionen einer Klasse	218
9.2	Überschreiben von Eigenschaften und Funktionen einer Klasse verhindern	221

9.3	Zugriff auf die Superklasse	221
9.4	Initialisierung und Vererbung	222
9.4.1	Zwei-Phasen-Initialisierung	223
9.4.2	Überschreiben von Initializern	230
9.4.3	Vererbung von Initializern	232
9.4.4	Required Initializer	233
10	Speicherverwaltung mit ARC	235
10.1	Strong Reference Cycles	239
10.1.1	Weak References	241
10.1.2	Unowned References	245
10.1.3	Weak Reference vs. Unowned Reference	248
11	Weiterführende Sprachmerkmale von Swift	249
11.1	Nested Types	249
11.2	Extensions	251
11.2.1	Computed Properties	251
11.2.2	Methoden	252
11.2.3	Initializer	253
11.2.4	Subscripts	255
11.2.5	Nested Types	256
11.3	Protokolle	256
11.3.1	Deklaration von Eigenschaften und Funktionen	258
11.3.2	Der Typ eines Protokolls	268
11.3.3	Protokolle und Extensions	270
11.3.4	Vererbung in Protokollen	274
11.3.5	Class-only-Protokolle	275
11.3.6	Optionale Eigenschaften und Funktionen	277
11.3.7	Protocol Composition	279
11.3.8	Delegation	281
12	Type Checking und Type Casting	285
12.1	Type Checking mit „is“	288
12.2	Type Casting mit „as“	289
13	Error Handling	291
13.1	Deklaration und Feuern eines Fehlers	291
13.2	Reaktion auf einen Fehler	295
13.2.1	Mögliche Fehler mittels do-catch auswerten	295
13.2.2	Mögliche Fehler in Optionals umwandeln	299
13.2.3	Mögliche Fehler weitergeben	299
13.2.4	Mögliche Fehler ignorieren	301

14	Generics	303
14.1	Generic Functions	304
14.2	Generic Types	307
14.3	Type Constraints	310
14.4	Associated Types	310
15	Dateien und Interfaces	315
15.1	Modules und Source Files	315
15.2	Access Control	316
15.2.1	Access Level	316
15.2.2	Explizite und implizite Zuweisung eines Access Levels	320
15.2.3	Besonderheiten	321
Teil II: Xcode		327
16	Grundlagen, Aufbau und Einstellungen von Xcode	329
16.1	Über Xcode	330
16.2	Arbeiten mit Xcode	331
16.2.1	Dateien und Formate eines Xcode-Projekts	331
16.2.2	Umgang mit Dateien und Ordnern	336
16.3	Der Aufbau von Xcode	341
16.3.1	Toolbar	341
16.3.2	Navigator	343
16.3.3	Editor	347
16.3.4	Inspectors	351
16.3.5	Debug Area	352
16.4	Einstellungen	352
16.4.1	General	352
16.4.2	Accounts	353
16.4.3	Behaviors	354
16.4.4	Navigation	355
16.4.5	Fonts & Colors	356
16.4.6	Text Editing	356
16.4.7	Key Bindings	357
16.4.8	Source Control	358
16.4.9	Components	358
16.4.10	Locations	359
16.4.11	Server & Bots	360
16.5	Projekteinstellungen	360
16.5.1	Einstellungen am Projekt	361
16.5.2	Einstellungen am Target	364
16.5.3	Einstellungen am Scheme	370

17	Arbeiten mit dem Interface Builder	375
17.1	Grundlegende Nutzung des Interface Builders	376
17.2	Interfaces mithilfe der Inspectors optimieren	377
17.2.1	Identity Inspector	378
17.2.2	Attributes Inspector	379
17.2.3	Size Inspector	380
17.2.4	Connections Inspector	381
17.3	Interface und Code koppeln	382
17.4	Arten von Interface-Dateien	385
17.4.1	Storyboards	385
17.4.2	XIB-Files	386
17.4.3	Neue Interface-Dateien erstellen	387
18	Dokumentation, Devices und Organizer	389
18.1	Dokumentation	389
18.1.1	Aufbau und Funktionsweise	390
18.1.2	Direktzugriff im Editor	393
18.2	Devices und Simulatoren	395
18.2.1	Simulatoren	397
18.2.2	Devices	399
18.3	Organizer	400
19	Debugging und Refactoring	403
19.1	Debugging	403
19.1.1	Konsolenausgaben	404
19.1.2	Arbeiten mit Breakpoints	405
19.1.3	Debug Navigator	410
19.2	Refactoring	411
19.3	Instruments	413
20	Tipps und Tricks für das effiziente Arbeiten mit Xcode	417
20.1	Code Snippets	417
20.2	Open Quickly	420
20.3	Related Items	421
20.4	Navigation über die Jump Bar	422
20.5	MARK, TODO und FIXME	422
20.6	Shortcuts für den Navigator	423
20.7	Clean Build	424

Teil III: Apple-Plattformen	425
21 macOS – Grundlagen	427
21.1 Über macOS	428
21.2 Funktionsweise einer macOS-App	429
21.2.1 Bestandteile einer macOS-App	429
21.2.2 App-Start	433
21.2.3 Das AppKit-Framework	436
21.2.4 Arten von macOS-Apps	436
21.3 Ein erstes macOS-Projekt	439
21.3.1 Auswahl einer Template-Vorlage	439
21.3.2 Rundgang durch die erstellten Dateien	443
21.3.3 Hello World	447
21.4 Der NSApplicationDelegate	452
21.5 NSWindow und NSWindowController im Detail	454
21.5.1 NSWindow	455
21.5.2 NSWindowController	456
21.5.3 Zusammenspiel	456
21.6 NSViewController im Detail	456
21.6.1 View life cycle	456
21.6.2 Einblenden neuer View-Controller	457
21.6.3 Ausblenden eines View-Controllers	465
21.6.4 Verknüpfen von View-Controllern mittels Segue	469
21.7 Oberflächen gestalten mit NSView	471
21.7.1 View-Hierarchien	471
21.7.2 Anpassen von Views im Attributes Inspector	474
21.7.3 Verfügbare NSView-Subklassen	475
21.7.4 Views und Actions	493
21.8 App-Icon	496
21.9 Target-Einstellungen	497
22 macOS – App-Entwicklung	501
22.1 Tabellen erstellen	501
22.1.1 Grundlegende Infos	502
22.1.2 Erstellen und Konfigurieren einer Table-View	504
22.1.3 Erstellen von Zellen	513
22.1.4 Zellen hinzufügen und entfernen	521
22.1.5 Auf Doppelklick reagieren	529
22.2 Menü anpassen	531
22.2.1 Hauptmenü über das Storyboard aktualisieren	533
22.2.2 Menüs im Code anpassen und aktualisieren	536
22.2.3 Tooltip definieren	540
22.2.4 Kontextmenü umsetzen	540

22.3	Alerts einblenden	542
22.3.1	Action-Buttons ergänzen	545
22.3.2	Hilfe-Button einbinden	547
22.3.3	Zusätzliche View in Alert einbinden	549
22.4	Touch Bar verwenden	550
22.4.1	Programmatische Bestandteile der Touch Bar	551
22.4.2	Touch Bar anpassen	551
22.4.3	Konfigurationsmöglichkeiten der Touch Bar	557
22.4.4	Touch Bar Items im Detail	563
22.4.5	Touch Bar im Interface Builder erzeugen	569
22.4.6	Prototyping im Playground	571
23	iOS – Grundlagen	573
23.1	Über iOS	574
23.2	Funktionsweise einer iOS-App	575
23.2.1	Bestandteile einer iOS-App	575
23.2.2	App-Start	577
23.2.3	Das UIKit-Framework	580
23.3	Ein erstes iOS-Projekt	581
23.3.1	Auswahl einer Template-Vorlage	581
23.3.2	Rundgang durch die erstellten Dateien	584
23.3.3	Hello World	590
23.4	Der UIApplicationDelegate	594
23.4.1	Lebenszyklus einer iOS-App	595
23.4.2	Die window-Property	598
23.4.3	Einsatzzweck des App Delegate	598
23.5	UIViewController im Detail	599
23.5.1	Aufbau	599
23.5.2	Ansicht eines View-Controllers anpassen	602
23.5.3	Verbindung zwischen Interface und Code	613
23.5.4	Lebenszyklus eines View-Controllers	634
23.5.5	Neuen View-Controller einblenden	636
23.6	Oberflächen gestalten mit UIView	658
23.6.1	Aufbau von Views	658
23.6.2	Erstellen von Views	659
23.6.3	Grundlegende Eigenschaften aller Views	663
23.6.4	Verfügbare UIView-Subklassen	670
23.6.5	Views mit Actions verbinden	686
23.7	Arbeit mit dem Simulator	689
23.7.1	Ausführen von Apps im Simulator	689
23.7.2	Arbeiten mit dem Simulator	692
23.7.3	Verwalten der Simulatoren	696
23.7.4	Einschränkungen des Simulators	698

23.8	App-Icon	699
23.9	Target-Einstellungen	702
24	iOS – App-Entwicklung	707
24.1	Aufbau einer Navigationsstruktur	707
24.1.1	UINavigationController im Code erstellen	708
24.1.2	UINavigationController im Storyboard erstellen	720
24.2	Erstellen einer Tab-Bar	733
24.2.1	Erstellen einer Tab-Bar im Code	733
24.2.2	Erstellen einer Tab-Bar im Storyboard	740
24.2.3	Der More-Tab	748
24.2.4	Zugriff auf zugrunde liegenden Tab-Bar-Controller	750
24.3	Erstellen von Tabellen	753
24.3.1	Funktionsweise einer Table-View	754
24.3.2	Hinzufügen einer Tabelle zu einem View-Controller	756
24.3.3	Erstellen von Zellen für eine Tabelle	756
24.3.4	Implementieren des Data Source	757
24.3.5	Wiederverwendung von Zellen	759
24.3.6	Table-View um weitere Bereiche ergänzen	761
24.3.7	Style einer Table-View verändern	764
24.3.8	Zellen im Storyboard gestalten	766
24.3.9	Größe einer Zelle verändern	776
24.3.10	Auf Auswahl einer Zelle reagieren	781
24.3.11	Der UITableViewController	788
24.3.12	Statische Tabellen im Storyboard erstellen	789
24.4	Eingabe von Text	794
24.4.1	Text eingeben über einfache Textfelder	794
24.4.2	Text eingeben und verwalten über umfangreiche Textansichten	805
24.4.3	Auf Ein- und Ausblenden des Keyboards reagieren	811
24.5	Einblenden von Alerts	815
24.5.1	Alert um Aktionen ergänzen	818
24.5.2	Alert um Textfelder ergänzen	822
24.6	Zugriff auf die Kamera und Fotos	824
24.6.1	Aufnahme und Auswahl auswerten	828
24.6.2	Videos aufnehmen und wiedergeben	832
24.6.3	Foto- und Videoaufnahme parallel erlauben	837
24.7	Erkennen von Gesten	841
24.7.1	UITapGestureRecognizer	842
24.7.2	UIPinchGestureRecognizer	845
24.7.3	UISwipeGestureRecognizer	848
24.7.4	UIPanGestureRecognizer	851
24.7.5	UILongPressGestureRecognizer	854
24.7.6	Weitere Gesture Recognizer	856
24.7.7	Erstellung im Code	857

25	watchOS – Grundlagen	859
25.1	Über watchOS	860
25.2	Funktionsweise einer watchOS-App	863
25.2.1	Bestandteile einer watchOS-App	863
25.2.2	App-Start	864
25.2.3	Das WatchKit-Framework	864
25.3	Ein erstes watchOS-Projekt	865
25.4	Die WatchKit App	869
25.4.1	Grundlegende Interface-Elemente	871
25.4.2	Oberflächen gestalten	877
25.4.3	Die Klasse WKInterfaceObject	888
25.5	Die WatchKit Extension	890
25.5.1	Der WKExtensionDelegate	891
25.5.2	Der WKInterfaceController	893
25.6	Grundlagen der App-Entwicklung	897
25.6.1	Interface und Code koppeln	897
25.6.2	Neuen Interface-Controller einblenden	904
25.6.3	context-Parameter	908
25.6.4	Aktuellen Interface-Controller ausblenden	914
25.6.5	Einschränkungen	915
25.7	App-Icon	916
26	watchOS – App-Entwicklung	917
26.1	Navigationsstruktur umsetzen	917
26.2	Alerts erstellen und anzeigen	922
26.3	Kontextmenü umsetzen	926
26.3.1	Kontextmenü im Storyboard umsetzen	927
26.3.2	Kontextmenü im Code umsetzen	932
26.3.3	Menu Items in Storyboard und Code mischen	935
26.3.4	Typische Einsatzzwecke von Kontextmenüs	936
26.4	Tabellen erstellen	936
26.4.1	Aufbau einer Tabelle	936
26.4.2	Row Controller-Klasse erstellen	938
26.4.3	Zellen erstellen und laden	939
26.4.4	Zellen konfigurieren	942
26.4.5	Tabelle mit verschiedenen Row Controllern umsetzen	945
26.4.6	Auf Zellenauswahl reagieren	950
26.4.7	Zellen hinzufügen und entfernen	956
26.4.8	Item Pagination	958
26.5	Text eingeben	960
26.5.1	Emojis in Texten verwenden	962
26.5.2	Mehrsprachige Empfehlungen umsetzen	965
26.6	Audio und Video wiedergeben	966

26.6.1	Medienwiedergabe über Interface-Element umsetzen	968
26.6.2	Medienwiedergabe ohne modales Interface umsetzen	971
26.7	Audio aufnehmen	973
26.8	Drehen der Digital Crown abfangen	976
26.9	Animationen durchführen	978
26.10	Komplikationen erstellen	979
26.10.1	Technische Funktionsweise von Komplikationen	980
26.10.2	Complication Families und Templates	981
26.10.3	Bestehendes Projekt um Complication-Support ergänzen	990
26.10.4	Neues watchOS-Projekt inklusive Complication-Support erstellen ..	998
26.10.5	Komplikationen für verschiedene Zeitpunkte konfigurieren	999
26.10.6	Data Provider	1001
26.10.7	Privatsphäre für Komplikationen definieren	1004
27	tvOS – Grundlagen und App-Entwicklung	1005
27.1	Über tvOS	1006
27.2	Funktionsweise einer tvOS-App	1008
27.3	Ein erstes tvOS-Projekt	1008
27.3.1	Auswahl einer Template-Vorlage	1008
27.3.2	Rundgang durch die erstellten Dateien	1011
27.3.3	Hello World	1012
27.4	App Delegate, View-Controller und Views	1015
27.5	Die Focus Engine im Detail	1015
27.5.1	Zu fokussierende View selbst definieren	1019
27.5.2	Fokus-Aktualisierung anstoßen	1022
27.6	Arbeiten mit dem Simulator	1024
27.7	App-Icon und Top Shelf Image	1026
28	Cross-Platform	1029
28.1	Das Foundation-Framework	1029
28.1.1	NSObject	1030
28.1.2	NSString	1030
28.1.3	NSNumber	1030
28.1.4	NSArray	1031
28.1.5	NSSet	1031
28.1.6	NSDictionary	1031
28.1.7	Mutable-Klassen	1031
28.2	MVC	1032
28.2.1	MVC in der Praxis	1033
28.2.2	Kommunikation zwischen Model und Controller	1034
28.2.3	Kommunikation zwischen View und Controller	1046
28.3	Auto Layout	1054
28.3.1	Grundlagen	1054

28.3.2	Constraints über Menü setzen	1055
28.3.3	Constraints durch Ziehen mit der Maus setzen	1058
28.3.4	Bestehende Constraints einsehen und bearbeiten	1059
28.3.5	Vorschau in Xcode anzeigen lassen	1061
28.3.6	Auto Layout deaktivieren	1063
28.4	Lokalisierung von Apps	1065
28.4.1	Grundlagen	1065
28.4.2	Interfaces übersetzen	1070
28.4.3	Verschiedene Sprachen einer App testen	1072
28.5	Asset Catalogs	1073
28.6	Nutzereinstellungen	1074

Teil IV: Frameworks und Technologien 1077

29 Authentifizierung 1079

29.1	Funktionsweise	1079
29.2	Verfügbare Authentifizierungstechniken	1081
29.3	Lokalisierung der Schaltflächen	1081
29.4	Zeitintervall zur Wiederverwendung von Touch ID	1082
29.5	Informationen zur Veränderung der biometrischen Nutzerdaten	1083
29.6	Authentifizierung abbrechen	1083
29.7	Mögliche Fehlercodes	1084
29.8	Tests im Simulator	1084

30 iCloud 1087

30.1	Nutzungsmöglichkeiten der iCloud	1087
30.2	Vorbereitung	1088
30.3	Nutzereinstellungen in der iCloud	1089
30.3.1	Nutzereinstellungen speichern und laden	1089
30.3.2	Auf Änderungen reagieren	1090
30.4	Zugriff auf iCloud Drive	1093
30.4.1	Container-Ordner in iCloud Drive erzeugen	1094
30.5	CloudKit	1097
30.5.1	Funktionsweise und Begrifflichkeiten	1098
30.5.2	Das CloudKit Dashboard	1099
30.5.3	Arbeiten mit dem CloudKit-Framework	1104

31 Siri 1111

31.1	Funktionsweise und Einschränkungen	1111
31.2	Siri-Support vorbereiten	1112
31.2.1	Siri-Capability aktivieren	1112
31.2.2	Info.plist aktualisieren	1113
31.2.3	Zugriff auf Siri erfragen	1114

31.3	Intents Extension	1116
31.3.1	Funktionsweise	1117
31.3.2	Unterstützte Intents definieren	1118
31.3.3	Intents implementieren	1124
31.3.4	Intents testen	1130
31.4	Intents UI Extension	1131
31.5	Siri Shortcuts	1136
31.5.1	Funktionsweise	1137
31.5.2	Siri Shortcuts mit NSUserActivity	1138
31.5.3	Siri Shortcuts mit Intents	1141
31.5.4	Shortcuts testen	1149
31.5.5	Shortcuts löschen	1152
Teil V: Source Control und Testing		1155
32	Source Control	1157
32.1	Basisfunktionen und -begriffe der Source Control	1157
32.2	Source Control in Xcode	1159
32.2.1	Bestehendes Projekt klonen	1160
32.2.2	Lokale Änderungen committen	1162
32.2.3	Lokale Änderungen verwerfen	1163
32.2.4	Pull und Push	1163
32.2.5	Aktuelle Branches vom Repository laden	1164
32.2.6	Git-Repository mit neuem Xcode-Projekt erzeugen	1164
32.2.7	Optische Source Control-Hervorhebungen im Editor	1165
32.2.8	Zugriff auf GitHub, GitLab und Bitbucket	1166
32.3	Version Editor	1167
33	Testing	1169
33.1	Unit-Tests	1169
33.1.1	Aufbau und Funktionsweise von Unit-Tests	1173
33.1.2	Aufbau einer Test-Case-Klasse	1175
33.1.3	Neue Test-Case-Klasse erstellen	1177
33.1.4	Ausführen von Unit-Tests	1178
33.1.5	Was sollte ich eigentlich testen?	1181
33.2	Performance-Tests	1181
33.3	UI-Tests	1183
33.3.1	Klassen für UI-Tests	1184
33.3.2	Aufbau von UI-Test-Klassen	1186
33.3.3	Automatisches Erstellen von UI-Tests	1186
33.3.4	Einsatz von UI-Tests	1187

Teil VI: Veröffentlichung von Apps	1189
34 Veröffentlichung im App Store	1191
34.1 Das Apple Developer Portal	1192
34.1.1 Zertifikate, App IDs und Provisioning Profiles	1195
34.1.2 Code Signing	1208
34.2 App Store Connect	1212
34.2.1 Apps für den App Store vorbereiten und verwalten	1213
34.2.2 Apps erstellen, hochladen und einreichen	1217
34.3 App Store Review Guidelines	1219
35 Das Business Model für Ihre App	1221
35.1 Geschäftsmodelle	1221
35.1.1 Free Model	1221
35.1.2 Freemium Model	1222
35.1.3 Subscription Model	1222
35.1.4 Paid Model	1223
35.1.5 Paymium Model	1223
35.2 App Bundles	1224
35.3 Universal Purchase für iOS und tvOS	1225
35.4 Veröffentlichung außerhalb des App Store	1226
35.4.1 Das Apple Developer Enterprise Program	1227
36 TestFlight	1229
36.1 TestFlight in App Store Connect	1229
36.2 TestFlight im App Store	1231
Index	1233