

Das Swift Handbuch

Apps programmieren für macOS,
iOS, watchOS und tvOS

DAS INHALTS- VERZEICHNIS

» Hier geht's
direkt
zum Buch

Inhalt

Vorwort	XXV
Teil I: Swift	1
1 Die Programmiersprache Swift	3
1.1 Die Geschichte von Swift	3
1.2 Die Bedeutung von Swift im Apple-Kosmos	5
1.3 Das UI-Framework: SwiftUI	5
1.4 Was Sie als App-Entwickler brauchen	6
1.5 Programmieren für Beginner (und darüber hinaus): Playgrounds	8
1.6 Weitere wichtige Ressourcen	10
1.6.1 Apple-Developer-App	11
1.6.2 Apples Developer-Website	11
1.6.3 Swift.org	12
1.6.4 In eigener Sache	13
2 Grundlagen der Programmierung	14
2.1 Grundlegendes	14
2.1.1 Swift Standard Library	14
2.1.2 print	16
2.1.3 Befehle und Semikolons	17
2.1.4 Operatoren	18
2.2 Variablen und Konstanten	20
2.2.1 Erstellen von Variablen und Konstanten	20

2.2.2	Variablen und Konstanten in der Konsole ausgeben	21
2.2.3	Type Annotation und Type Inference	22
2.2.4	Gleichzeitiges Erstellen und Deklarieren mehrerer Variablen und Konstanten	24
2.2.5	Namensrichtlinien	25
2.3	Kommentare	25
3	Schleifen und Abfragen	27
3.1	Schleifen	27
3.1.1	For-In	27
3.1.2	While	29
3.1.3	Repeat-While	31
3.2	Abfragen	32
3.2.1	If	32
3.2.2	Switch	37
3.2.3	Guard	41
3.3	Control Transfer Statements	43
3.3.1	Anstoßen eines neuen Schleifendurchlaufs mit continue	43
3.3.2	Verlassen der kompletten Schleife mit break	43
3.3.3	Labeled Statements	44
4	Typen in Swift	47
4.1	Integer	49
4.2	Fließkommazahlen	50
4.3	Bool	51
4.4	String	51
4.4.1	Erstellen eines Strings	52
4.4.2	Zusammenfügen von Strings	52
4.4.3	Character auslesen	54
4.4.4	Character mittels Index auslesen	54
4.4.5	Character entfernen und hinzufügen	56
4.4.6	Anzahl der Character zählen	58
4.4.7	Präfix und Suffix prüfen	58
4.4.8	String Interpolation	58
4.5	Array	59
4.5.1	Erstellen eines Arrays	60

4.5.2	Zusammenfügen von Arrays	61
4.5.3	Inhalte eines Arrays leeren	62
4.5.4	Prüfen, ob ein Array leer ist	62
4.5.5	Anzahl der Elemente eines Arrays zählen	63
4.5.6	Zugriff auf die Elemente eines Arrays	63
4.5.7	Neue Elemente zu einem Array hinzufügen	64
4.5.8	Bestehende Elemente aus einem Array entfernen	65
4.5.9	Bestehende Elemente eines Arrays ersetzen	66
4.5.10	Alle Elemente eines Arrays auslesen und durchlaufen	67
4.6	Set	68
4.6.1	Erstellen eines Sets	68
4.6.2	Inhalte eines bestehenden Sets leeren	69
4.6.3	Prüfen, ob ein Set leer ist	70
4.6.4	Anzahl der Elemente eines Sets zählen	70
4.6.5	Element zu einem Set hinzufügen	70
4.6.6	Element aus einem Set entfernen	71
4.6.7	Prüfen, ob ein bestimmtes Element in einem Set vorhanden ist ..	71
4.6.8	Alle Elemente eines Sets auslesen und durchlaufen	72
4.6.9	Sets miteinander vergleichen	72
4.6.10	Neue Sets aus bestehenden Sets erstellen	75
4.7	Dictionary	77
4.7.1	Erstellen eines Dictionaries	77
4.7.2	Prüfen, ob ein Dictionary leer ist	79
4.7.3	Anzahl der Schlüssel-Wert-Paare eines Dictionaries zählen	79
4.7.4	Wert zu einem Schlüssel eines Dictionaries auslesen	79
4.7.5	Neues Schlüssel-Wert-Paar zu Dictionary hinzufügen	80
4.7.6	Bestehendes Schlüssel-Wert-Paar aus Dictionary entfernen	81
4.7.7	Bestehendes Schlüssel-Wert-Paar aus Dictionary verändern	81
4.7.8	Alle Schlüssel-Wert-Paare eines Dictionaries auslesen und durchlaufen	82
4.8	Tuple	83
4.8.1	Zugriff auf die einzelnen Elemente eines Tuples	84
4.8.2	Tuple und switch	85
4.9	Optional	89
4.9.1	Deklaration eines Optionals	89

4.9.2	Zugriff auf den Wert eines Optionals	90
4.9.3	Optional Binding	92
4.9.4	Implicitly Unwrapped Optional	95
4.9.5	Optional Chaining	96
4.9.6	Optional Chaining über mehrere Eigenschaften und Funktionen	101
4.10	Any und AnyObject	106
4.11	Type Alias	106
4.12	Value Type versus Reference Type	107
4.12.1	Reference Types auf Gleichheit prüfen	109
5	Funktionen	111
5.1	Funktionen mit Parametern	112
5.1.1	Argument Labels und Parameter Names	114
5.1.2	Default Value für Parameter	116
5.1.3	Variadic Parameter	118
5.1.4	In-Out-Parameter	119
5.2	Funktionen mit Rückgabewert	120
5.3	Function Types	122
5.3.1	Funktionen als Variablen und Konstanten	124
5.4	Verschachtelte Funktionen	125
5.5	Closures	126
5.5.1	Closures als Parameter von Funktionen	127
5.5.1.1	Implicit Return	130
5.5.1.2	Shorthand Argument Names	130
5.5.2	Trailing Closures	131
5.5.3	Autoclosures	132
6	Enumerations, Structures und Classes	134
6.1	Enumerations	134
6.1.1	Enumerations und switch	137
6.1.2	Associated Values	138
6.1.3	Raw Values	141
6.2	Structures	143
6.2.1	Erstellen von Structures und Instanzen	143
6.2.2	Eigenschaften und Funktionen	144
6.2.2.1	Memberwise Initializer	148

6.3	Classes	151
6.3.1	Erstellen von Klassen und Instanzen	151
6.3.2	Eigenschaften und Funktionen	152
6.4	Enumeration vs. Structure vs. Class	154
6.4.1	Gemeinsamkeiten und Unterschiede	154
6.4.2	Wann nimmt man was?	155
6.4.2.1	Enumeration	156
6.4.2.2	Structure	156
6.4.2.3	Class	156
6.5	self	157
7	Eigenschaften und Funktionen von Typen	160
7.1	Properties	160
7.1.1	Stored Property	161
7.1.2	Lazy Stored Property	164
7.1.2.1	Einsatzzweck von Lazy Stored Properties	167
7.1.3	Computed Property	168
7.1.4	Read-Only Computed Property	171
7.1.5	Property Observer	173
7.1.6	Property Wrapper	177
7.1.6.1	Standardwerte festlegen	179
7.1.6.2	Weitere Parameter ergänzen	180
7.1.6.3	Projected Value	182
7.1.7	Type Property	183
7.2	Globale und lokale Variablen	186
7.3	Methoden	189
7.3.1	Instance Methods	189
7.3.1.1	mutating	191
7.3.2	Type Methods	192
7.4	Subscripts	193
8	Initialisierung	199
8.1	Aufgabe der Initialisierung	200
8.2	Erstellen eigener Initializer	201
8.3	Initializer Delegation	208
8.3.1	Initializer Delegation bei Value Types	208

8.3.2	Initializer Delegation bei Reference Types	209
8.4	Failable Initializer	212
8.5	Required Initializer	215
8.6	Deinitialisierung	216
9	Vererbung	218
9.1	Überschreiben von Eigenschaften und Funktionen einer Klasse	222
9.2	Überschreiben von Eigenschaften und Funktionen einer Klasse verhindern	224
9.3	Zugriff auf die Superklasse	225
9.4	Initialisierung und Vererbung	226
9.4.1	Zwei-Phasen-Initialisierung	227
9.4.2	Überschreiben von Initializern	234
9.4.3	Vererbung von Initializern	237
9.4.4	Required Initializer	237
10	Speicherverwaltung mit ARC	239
10.1	Strong Reference Cycles	243
10.1.1	Weak References	245
10.1.2	Unowned References	249
10.1.3	Weak Reference vs. Unowned Reference	251
11	Weiterführende Sprachmerkmale von Swift	252
11.1	Nested Types	252
11.2	Extensions	254
11.2.1	Computed Properties	255
11.2.2	Methoden	255
11.2.3	Initializer	256
11.2.4	Subscripts	259
11.2.5	Nested Types	259
11.3	Protokolle	260
11.3.1	Deklaration von Eigenschaften und Funktionen	262
11.3.1.1	Properties	262
11.3.1.2	Methoden	264
11.3.1.3	Subscripts	267
11.3.1.4	Initializer	268

11.3.2	Der Typ eines Protokolls	272
11.3.3	Protokolle und Extensions	275
11.3.3.1	Bestehenden Typ mittels Extension um Protokoll ergänzen	275
11.3.3.2	Protokoll mittels Extension um neue Funktionen und Standardimplementierung erweitern	277
11.3.4	Vererbung in Protokollen	280
11.3.5	Class-only-Protokolle	281
11.3.6	Optionale Eigenschaften und Funktionen	282
11.3.6.1	Umgang mit Protokoll-Type	284
11.3.7	Protocol Composition	285
11.3.8	Delegation	286
11.3.9	Übersicht diverser vorhandener Protokolle	289
11.3.9.1	Hashable	289
11.3.9.2	Identifiable	291
11.4	Key-Path	291
12	Type Checking und Type Casting	295
12.1	Type Checking mit „is“	298
12.2	Type Casting mit „as“	299
13	Error Handling	302
13.1	Deklaration und Feuern eines Fehlers	303
13.2	Reaktion auf einen Fehler	306
13.2.1	Mögliche Fehler mittels do-catch auswerten	307
13.2.2	Mögliche Fehler in Optionals umwandeln	311
13.2.3	Mögliche Fehler weitergeben	311
13.2.4	Mögliche Fehler ignorieren	313
14	Generics	314
14.1	Generic Functions	315
14.2	Generic Types	319
14.3	Type Constraints	321
14.4	Associated Types	322

15	Nebenläufigkeit	327
15.1	Asynchronen Code schreiben und aufrufen	327
15.2	Mehrere asynchrone Funktionen parallel ausführen	331
15.3	Actors	332
16	Dateien und Interfaces	335
16.1	Modules und Source Files	335
16.2	Access Control	336
16.2.1	Access Level	336
16.2.1.1	Private Access	337
16.2.1.2	File-private Access	337
16.2.1.3	Internal Access	338
16.2.1.4	Public Access	339
16.2.1.5	Open Access	339
16.2.1.6	Zusammenfassung und Übersicht	339
16.2.2	Explizite und implizite Zuweisung eines Access Levels	340
16.2.3	Besonderheiten	342
16.2.3.1	Variablen und Konstanten	342
16.2.3.2	Tuples	342
16.2.3.3	Type Aliase	343
16.2.3.4	Funktionen	343
16.2.3.5	Enumerations	344
16.2.3.6	Properties	344
16.2.3.7	Subscripts	344
16.2.3.8	Getter und Setter	344
16.2.3.9	Initializer	345
16.2.3.10	Vererbung	346
16.2.3.11	Extensions	347
16.2.3.12	Protokolle	347
	Teil II: Xcode	349
17	Grundlagen, Aufbau und Einstellungen von Xcode	351
17.1	Über Xcode	352
17.2	Arbeiten mit Xcode	353
17.2.1	Dateien und Formate eines Xcode-Projekts	354

17.2.1.1	Projekte	354
17.2.1.2	Targets	358
17.2.1.3	Schemes	359
17.2.2	Umgang mit Dateien und Ordnern	359
17.2.2.1	Dateien in Xcode hinzufügen	360
17.2.2.2	Ordner in Xcode hinzufügen	361
17.2.2.3	Bereits vorhandene Dateien einem Xcode-Projekt hinzufügen	361
17.2.2.4	Dateien und Ordner löschen	364
17.3	Der Aufbau von Xcode	365
17.3.1	Toolbar	365
17.3.2	Navigator	368
17.3.3	Editor	373
17.3.4	Inspectors	378
17.3.5	Debug Area	381
17.4	Einstellungen	382
17.4.1	General	382
17.4.2	Accounts	383
17.4.3	Behaviors	385
17.4.4	Navigation	386
17.4.5	Themes	387
17.4.6	Text Editing	388
17.4.7	Key Bindings	389
17.4.8	Source Control	390
17.4.9	Platforms	391
17.4.10	Locations	392
17.5	Projekteinstellungen	393
17.5.1	Einstellungen am Projekt	394
17.5.2	Einstellungen am Target	397
17.5.2.1	General	397
17.5.2.2	Signing & Capabilities	398
17.5.2.3	Resource Tags	399
17.5.2.4	Info	401
17.5.2.5	Build Settings	402
17.5.2.6	Build Phases	402

17.5.2.7	Build Rules	403
17.5.3	Einstellungen am Scheme	404
17.5.3.1	Neues Scheme erstellen	406
17.5.3.2	Schemes verwalten	407
17.5.3.3	Ausführungsmöglichkeiten eines Schemes	408
18	Dokumentation, Devices und Organizer	409
18.1	Dokumentation	409
18.1.1	Aufbau und Funktionsweise	410
18.1.2	Direktzugriff im Editor	413
18.2	Devices und Simulatoren	415
18.2.1	Simulatoren	417
18.2.2	Devices	419
18.3	Organizer	421
19	Debugging und Refactoring	423
19.1	Debugging	423
19.1.1	Konsolenausgaben	425
19.1.2	Arbeiten mit Breakpoints	426
19.1.2.1	Breakpoints setzen und aktivieren	426
19.1.2.2	Variables View einsetzen	426
19.1.2.3	Ausführung der App fortsetzen	428
19.1.2.4	Breakpoints konfigurieren	428
19.1.2.5	Breakpoints vollständig deaktivieren	429
19.1.2.6	Breakpoint Navigator	430
19.1.3	Debug Navigator	431
19.2	Refactoring	433
19.3	Instruments	435
20	Tipps und Tricks für das effiziente Arbeiten mit Xcode	439
20.1	Code Snippets	439
20.2	Open Quickly	442
20.3	Related Items	442
20.4	Navigation über die Jump Bar	444
20.5	MARK, TODO und FIXME	445
20.6	Shortcuts für den Navigator	446

20.7	Clean Build	446
20.8	Playgrounds	446
20.8.1	Was sind Playgrounds?	447
20.8.2	Code schreiben und testen	449
20.8.3	Dateien hinzufügen	454
20.8.4	Kommentare und Dokumentation	456
20.8.5	Swift Playgrounds-App	459
Teil III: App-Entwicklung		463
21	Grundlagen der App-Entwicklung	465
21.1	Die Basis: SwiftUI	465
21.2	Bestandteile einer App	467
21.2.1	Umsetzung der Daten	468
21.2.2	Umsetzung der Ansichten	468
21.2.3	Weitere Frameworks	468
21.3	Die Syntax von SwiftUI	469
21.4	Aufbau einer App	470
21.5	Das View-Protokoll	471
21.6	Aktualisierung von Views mittels Status	472
21.7	Grundlagen des Status	474
21.8	Anpassung von Views mittels Modifier	477
21.9	Gruppierung von Views mittels Containern	480
21.10	Praxis: Unsere erste App	482
21.10.1	Bestandteile des neuen Projekts	488
21.10.2	Änderung des Textes	491
21.10.3	Einsatz der Preview	492
22	Views in SwiftUI	494
22.1	Textdarstellung und -bearbeitung	494
22.1.1	Text	494
22.1.1.1	Anpassung der Schriftart	496
22.1.1.2	Formatierung von Texten	501
22.1.1.3	Übersetzung von Texten	503
22.1.1.4	Umgang mit Datumsangaben	504
22.1.2	TextField	504

22.1.3	SecureField	509
22.1.4	TextEditor	510
22.1.4.1	Formatierung des Textes	511
22.2	Bilder	513
22.2.1	Image-Instanz erstellen	515
22.2.2	Größe einer Image-Instanz ändern	518
22.3	Schaltflächen	521
22.3.1	Button	521
22.3.2	EditButton	526
22.3.3	PasteButton	527
22.4	Werteauswahl	529
22.4.1	Toggle	529
22.4.2	Slider	534
22.4.3	Stepper	541
22.4.4	Picker	547
22.4.5	DatePicker	550
22.4.6	MultiDatePicker	555
22.4.7	ColorPicker	557
22.5	Weitere Views	561
22.5.1	Label	561
22.5.2	ProgressView	564
22.5.3	Gauge	568
23	View-Layout	573
23.1	Stacks	573
23.1.1	HStack	574
23.1.2	VStack	578
23.1.3	ZStack	581
23.1.4	Stacks verschachteln	584
23.1.5	Lazy Stacks	585
23.2	Listen	589
23.2.1	List	589
23.2.2	ForEach	612
23.3	Grids	625
23.3.1	Grid	625

23.3.2	LazyHGrid und LazyVGrid	627
23.4	Table	643
23.4.1	Zellen selektieren	644
23.4.2	Sortierung ändern.....	645
23.5	Container-Views	647
23.5.1	Form.....	647
23.5.2	Section.....	649
23.5.3	Group.....	652
23.5.4	GroupBox	658
23.5.5	ViewThatFits.....	661
23.6	Weitere Views	663
23.6.1	ScrollView.....	663
23.6.2	OutlineGroup.....	668
23.6.3	DisclosureGroup	673
23.6.4	Spacer.....	679
23.6.5	Divider.....	683
24	Navigation	685
24.1	NavigationStack und NavigationLink	685
24.1.1	Titel in Navigation-Bar setzen	690
24.1.2	Eigene View zur Darstellung eines NavigationLink nutzen	694
24.1.3	Anzeige einer Ziel-View auf Basis von Daten.....	696
24.1.4	Programmatische Steuerung des Navigation-Stacks	700
24.2	NavigationSplitView.....	704
24.2.1	Verknüpfung von NavigationSplitView und List.....	706
24.2.2	Sichtbarkeit der Spalten steuern.....	708
24.2.3	Breite der Spalten anpassen.....	709
24.2.4	Verhalten von NavigationSplitView unter den verschiedenen Apple-Plattformen	710
24.3	TabView.....	716
24.4	HSplitView und VSplitView	724
24.5	Funktionen zur Präsentation von Views.....	725
24.5.1	Sheet einblenden.....	725
24.5.2	View über gesamtes Fenster legen	736
24.5.3	Popover einblenden	741

25	Weitere View-Konfigurationen	748
25.1	Toolbar	748
25.2	Alerts	758
25.3	Confirmation Dialog	761
25.4	Farben	763
25.5	View-Events	765
26	Status	767
26.1	Property	769
26.2	State	771
26.3	Binding	773
26.4	ObservedObject	782
26.4.1	Datenmodell vorbereiten	783
26.4.2	Datenmodell in View einbinden	784
26.4.3	Auf Änderungen reagieren	788
26.5	StateObject	791
26.6	EnvironmentObject	793
26.7	@Observable-Makro und Bindable	799
26.8	Environment	801
26.9	SceneStorage	807
26.10	AppStorage	810
26.11	Source of Truth vs. Derived Value	811
26.12	Best Practices	815
27	Datenhaltung	821
27.1	UserDefaults	821
27.1.1	UserDefaults und SwiftUI	823
27.2	SwiftData	824
27.2.1	Grundlegende Funktionsweise von SwiftData	824
27.2.2	Erstellen des eigenen Datenmodells	825
27.2.3	Model-Container erzeugen	827
27.2.4	Elemente im ModelContext verwalten	828
27.3	Core Data	833
27.3.1	Grundlegende Funktionsweise von Core Data	834
27.3.2	Grundlegende Elemente beim Einsatz von Core Data	834

27.3.2.1	NSPersistentStore	835
27.3.2.2	NSManagedObjectModel	835
27.3.2.3	NSManagedObject	836
27.3.2.4	NSManagedObjectContext	836
27.3.2.5	NSPersistentStoreCoordinator	836
27.3.2.6	NSPersistentContainer	837
27.3.3	Einen Core Data Stack erstellen	837
27.3.4	Ein Managed Object Model erstellen	839
27.3.4.1	Entitäten und Eigenschaften erstellen	840
27.3.4.2	Relationships zwischen Entitäten	843
27.3.4.3	Entitäten und Attribute als Managed Objects	847
27.3.5	Grundlegende Core-Data-Operationen	852
27.3.5.1	Managed Object erstellen und konfigurieren	852
27.3.5.2	Gespeicherte Managed Objects laden	853
27.3.5.3	Bestehende Managed Objects löschen	854
27.3.6	Core Data mit SwiftUI	854
27.3.6.1	NSManagedObject als Status	854
27.3.6.2	Zugriff auf Managed Object Context	856
27.3.6.3	Auslesen persistent gespeicherter Elemente	858
28	Weitere Projektkonfigurationen	862
28.1	Cross-Platform-Entwicklung	862
28.1.1	Neue Targets hinzufügen	863
28.1.2	Target-Zuweisung	866
28.1.3	Plattform im Code prüfen	868
28.1.4	Funktionen auf Verfügbarkeit prüfen	869
28.2	Mehrsprachigkeit	871
28.2.1	Grundlagen	871
28.2.1.1	Erstellen eines String Catalogs	872
28.2.1.2	Übersetzungen im Code kennzeichnen	874
28.2.1.3	Auf Übersetzungen im Code zugreifen	876
28.2.1.4	Übersetzungen mit dynamischem Parameter konfigurieren	876
28.2.2	Verschiedene Sprachen einer App testen	880
28.3	Asset Catalogs	881

29	Preview und Library	886
29.1	Preview	886
29.1.1	Funktionsweise der Preview	888
29.1.2	Konfiguration der Preview	888
29.1.3	Preview ausführen und anhalten	890
29.2	Library	892
29.3	Attributes Inspector	894
Teil IV: Source Control und Testing		897
30	Source Control	899
30.1	Basisfunktionen und -begriffe der Source Control	899
30.2	Source Control in Xcode	901
30.2.1	Bestehendes Projekt klonen	903
30.2.2	Lokale Änderungen committen	905
30.2.3	Lokale Änderungen verwerfen	908
30.2.4	Pull und Push	909
30.2.5	Aktuelle Branches vom Repository laden	910
30.2.6	Git-Repository mit neuem Xcode-Projekt erzeugen	910
30.2.7	Optische Source-Control-Hervorhebungen im Editor	911
30.2.8	Zugriff auf GitHub, GitLab und Bitbucket	912
30.3	Source Control Navigator	913
30.4	Code Review-Mode	914
31	Testing	917
31.1	Unit-Tests	917
31.1.1	Aufbau und Funktionsweise von Unit-Tests	922
31.1.2	Aufbau einer Test-Case-Klasse	925
31.1.3	Neue Test-Case-Klasse erstellen	927
31.1.4	Ausführen von Unit-Tests	928
31.1.5	Was sollte ich eigentlich testen?	930
31.2	Performancetests	931
31.3	UI-Tests	933
31.3.1	Klassen für UI-Tests	934
31.3.2	Aufbau von UI-Test-Klassen	937
31.3.3	Automatisches Erstellen von UI-Tests	937
31.3.4	Einsatz von UI-Tests	938

Teil V: Veröffentlichung von Apps	939
32 Veröffentlichung im App Store	941
32.1 Das Apple Developer Portal	942
32.1.1 Zertifikate, App IDs und Provisioning Profiles	946
32.1.1.1 Erstellen von Entwicklerzertifikaten	948
32.1.1.2 Erstellen von App IDs	952
32.1.1.3 Hinzufügen von Devices	954
32.1.1.4 Erstellen von Provisioning Profiles	957
32.1.2 Code Signing	964
32.1.2.1 Automatic Code Signing	965
32.1.2.2 Manual Code Signing	966
32.1.2.3 Code Signing in den Build Settings	967
32.2 App Store Connect	969
32.2.1 Apps für den App Store vorbereiten und verwalten	970
32.2.2 Apps erstellen, hochladen und einreichen	974
32.3 App Store Review Guidelines	977
33 Das Business Model für Ihre App	979
33.1 Geschäftsmodelle	979
33.1.1 Free Model	980
33.1.2 Freemium Model	980
33.1.3 Subscription Model	980
33.1.4 Paid Model	981
33.1.5 Paymium Model	982
33.2 App Bundles	982
33.3 Veröffentlichung außerhalb des App Store	984
33.3.1 Das Apple Developer Enterprise Program	985
34 TestFlight	986
34.1 TestFlight in App Store Connect	986
34.2 TestFlight im App Store	988
Index	991