

Inhalt

Vorwort	XV
1 Über iOS	1
1.1 Was ist iOS?	1
1.1.1 iOS und OS X	2
1.1.2 Besonderheiten der iOS-Plattform	3
1.2 iOS für Entwickler	4
1.2.1 Hardware für Entwickler	4
1.2.2 Software für Entwickler	6
1.2.3 Das Apple Developer Program	6
1.3 Der Aufbau von iOS	8
1.3.1 Die vier Schichten von iOS	8
1.4 Die perfekte iOS-App	10
1.4.1 Apple Human Interface Guidelines	11
2 Die Programmiersprache – Objective-C	13
2.1 Über Objective-C und objektorientierte Programmierung	13
2.2 Grundlagen der Programmierung	14
2.2.1 Objekte	15
2.2.2 Primitive Datentypen	15
2.2.3 Variablen	16
2.2.4 Operatoren	17
2.2.5 Abfragen und Schleifen	18
2.2.6 Kommentare	22
2.3 Aufbau einer Klasse	23
2.3.1 Grundlagen	23
2.3.2 Die Header-Datei	23
2.3.3 Die Implementation-Datei	25
2.3.4 Los geht's: Unsere erste Klasse!	26
2.4 Methoden	29
2.4.1 Grundlagen	29
2.4.2 Aufbau von Methoden	29

2.4.3 Methoden in Header- und Implementation-Dateien einer Klasse	31
2.4.4 Implementierung von Methoden	32
2.4.5 Methoden aufrufen	34
2.4.6 Klassen- und Instanzmethoden	36
2.5 Instanzvariablen	37
2.6 Properties	38
2.6.1 Grundlagen	38
2.6.2 Aufbau einer Property	38
2.6.3 Setter und Getter	39
2.6.4 Die Punktnotation	40
2.6.5 Optionen	41
2.6.6 Direktzugriff auf Properties	42
2.6.7 Setter und Getter überschreiben	44
2.7 Namenskonventionen	46
2.7.1 Grundlagen	46
2.7.2 Klassen	47
2.7.3 Methoden	47
2.7.4 Properties	47
2.8 Vererbung	47
2.8.1 Methoden der Superklasse überschreiben	50
2.9 Kategorien	51
2.9.1 Aufbau von Kategorien	51
2.9.2 Kategorien in Xcode erstellen	52
2.10 Erweiterungen	54
2.10.1 Grundlagen	54
2.10.2 Aufbau von Erweiterungen	54
2.10.3 Erweiterungen innerhalb der Implementation-Datei	55
2.10.4 Erweiterungen in Xcode erstellen	56
2.11 Protokolle	57
2.11.1 Grundlagen	57
2.11.2 Aufbau von Protokollen	58
2.11.3 Zuweisen eines Protokolls zu einer Klasse	58
2.11.4 Vererbung in Protokollen	59
2.11.5 Protokolle in Xcode erstellen	59
2.12 #import und @class	61
2.12.1 #import	61
2.12.2 @class	62
2.13 Blöcke	63
2.13.1 Ja zu Blöcken!	63
2.13.2 Was sind Blöcke?	64
2.13.3 Aufbau eines Blocks	64
2.13.4 Blockvariablen	68
2.13.5 Globale Blöcke	68

3 Programmierung für iOS	71
3.1 Grundlagen	71
3.2 Foundation Framework	72
3.2.1 Grundlagen	72
3.2.2 Die wichtigsten Klassen aus dem Foundation Framework und ihre Funktionen	72
3.3 UIKit Framework	75
3.4 Instanziierung von Objekten	76
3.4.1 Grundlagen	76
3.4.2 alloc und init	76
3.4.3 Zeiger	78
3.5 Speicherverwaltung mit ARC	79
3.6 init im Detail	83
3.6.1 Grundlagen	83
3.6.2 Erstellen mehrerer init-Methoden	84
3.6.3 Designated initializer	85
3.7 Kurzschreibweisen zum Erstellen von Objekten	87
3.7.1 Grundlagen	87
3.7.2 NSString	88
3.7.3 NSArray	88
3.7.4 NSDictionary	89
3.7.5 NSNumber	89
3.7.6 Mutable-Klassen werden nicht unterstützt	90
3.8 Singletons	90
3.8.1 Grundlagen	90
3.9 Konstanten	92
3.9.1 Grundlagen	92
3.9.2 Deklaration von Konstanten	93
3.10 Strukturen	94
3.10.1 Grundlagen	94
3.10.2 enum	94
3.10.3 typedef	95
3.11 Doxygen-Dokumentation	96
3.11.1 Grundlagen	96
3.11.2 Besonderheiten bei Methoden	97
3.11.3 Doxygen-Dokumentation in Xcode	98
3.12 Nebenläufigkeit mit Grand Central Dispatch	99
3.12.1 Grundlagen	99
3.12.2 Parallel laufenden Code erstellen	100
3.13 Grundlegende Struktur einer App	101
3.13.1 Grundlagen	101
3.13.2 main.m	101

3.13.3 Info.plist	102
3.13.4 App Delegate	102
3.14 Lebenszyklus einer iOS-App	103
3.14.1 Start einer App	103
3.14.2 Lebenszyklus einer App	104
3.14.3 Die Methoden des App Delegate	105
3.14.4 Start der App	106
3.15 Tipps für die tägliche Arbeit	108
3.15.1 Die netten Kleinigkeiten	108
3.15.2 Fast enumeration	108
3.15.3 Type casting	109
3.15.4 Xcode-Beispielprojekte	109
4 Der Neue im Club – Swift	111
4.1 Programmierst du noch oder swifst du schon?	111
4.1.1 Über Swift	111
4.1.2 Voraussetzungen zur Nutzung von Swift	111
4.1.3 Swift und Objective-C	112
4.1.4 Playgrounds	112
4.1.5 Grundlegendes zu Swift	114
4.2 Variablen und Konstanten	114
4.2.1 Grundlagen	114
4.2.2 Typzuweisung bei Variablen und Konstanten	115
4.2.3 Typkonvertierung in Swift	116
4.2.4 Arrays und Dictionaries	117
4.3 Abfragen und Schleifen	118
4.4 Funktionen	120
4.4.1 Eine erste einfache Swift-Funktion	120
4.4.2 Funktionen mit Rückgabewert	120
4.4.3 Funktionen mit Parametern	121
4.4.4 Funktionen mit mehreren Parametern als Array	122
4.4.5 Verschachtelte Funktionen	122
4.4.6 Funktionen als Rückgabewert	123
4.5 Klassen und Objekte	124
4.5.1 Grundlagen	124
4.5.2 Aufbau einer Klasse	124
4.5.3 Erstellen einer neuen Instanz einer Klasse	124
4.5.4 Eigene Initialisierer erstellen	125
4.5.5 Vererbung in Swift	126
4.6 Wie geht es weiter?	126

5 Die Entwicklungsumgebung – Xcode	127
5.1 Willkommen bei Xcode!	127
5.1.1 Was ist Xcode?	128
5.1.2 Interface Builder und Xcode – endlich vereint!	128
5.2 Arbeiten mit Xcode	129
5.2.1 Dateien und Formate eines Xcode-Projekts	129
5.2.2 Umgang mit Dateien und Ordnern in Xcode	133
5.3 Der Aufbau von Xcode	136
5.3.1 Die Toolbar	137
5.3.2 Die Navigation Area	138
5.3.3 Die Editor Area	142
5.3.4 Die Utilities Area	144
5.3.5 Die Debug Area	145
5.4 Einstellungen in Xcode	146
5.4.1 Anpassen von Xcode	146
5.4.2 General	146
5.4.3 Accounts	147
5.4.4 Behaviors	148
5.4.5 Navigation	148
5.4.6 Fonts & Colors	149
5.4.7 Text Editing	150
5.4.8 Key Bindings	150
5.4.9 Source Control	151
5.4.10 Downloads	152
5.4.11 Locations	152
5.5 Projekteinstellungen	153
5.5.1 Grundlagen	153
5.5.2 Einstellungen am Projekt	155
5.5.3 Einstellungen am Target	158
5.5.4 Einstellungen am Scheme	164
5.6 Grafiken und Asset-Bundles	166
5.7 Lokalisierung mit Localizable.strings	168
5.7.1 Grundlagen	168
5.7.2 NSLocalizedString	168
5.7.3 Erstellen der Localizable.strings-Datei	169
5.7.4 Localized String mit Parameter	171
5.7.5 Alle Localized Strings automatisch auslesen	172
5.8 Der iOS-Simulator	173
5.8.1 Grundlagen	173
5.8.2 Funktionen und Möglichkeiten des Simulators	174
5.8.3 Performance und Einschränkungen des Simulators	176
5.9 Dokumentation	177
5.9.1 Nichts geht über die Dokumentation!	177

5.9.2	Das Documentation-Window	179
-	5.9.3 Direktes Aufrufen der Dokumentation aus Xcode heraus	182
5.10	Devices	183
5.11	Organizer	185
5.12	Debugging in Xcode	187
5.12.1	Grundlagen – über das Debugging	187
5.12.2	Die Debug Area	188
5.12.3	Die Arbeit mit dem Debugger - NSLog und Breakpoints	189
5.12.4	Debug Navigator	197
5.13	Refactoring	198
5.13.1	Grundlagen	198
5.13.2	Refactoring-Funktionen in Xcode	199
5.14	Instruments	201
5.14.1	Über Instruments	201
5.14.2	Nächste Schritte	204
5.15	Tipps für die tägliche Arbeit	204
5.15.1	Man lernt immer was dazu!	204
5.15.2	Code Snippets	204
5.15.3	Open Quickly	206
5.15.4	Caller einer Methode feststellen	206
5.15.5	Speicherorte für Ordner und Dateien ändern	207
5.15.6	Shortcuts für die Navigation Area	208
5.15.7	Snapshots erstellen und zurückspielen	208
5.15.8	Clean Build	209
6	MVC – Model-View-Controller	211
6.1	MVC ... was?	211
6.2	MVC in der Praxis	213
6.3	Kommunikation zwischen Model und Controller	213
6.3.1	Key-Value-Observing	214
6.3.2	Notifications	218
6.4	Kommunikation zwischen View und Controller	220
6.4.1	Target-Action	220
6.4.2	Delegates	222
7	Die Vielfalt der (View-)Controller	225
7.1	Alles beginnt mit einem View-Controller	225
7.2	UIViewController – die Mutter aller View-Controller	227
7.2.1	Grundlagen	227
7.2.2	Wichtige Methoden von UIViewController	228
7.2.3	UIView – fester Bestandteil eines jeden UIViewControllers	231
7.3	View-Controller-Hierarchien	232

7.4	View-Controller erstellen mit dem Interface Builder	234
7.4.1	Grundlagen	234
7.4.2	View-Controller mit Nib-File	234
7.5	Storyboards	263
7.5.1	Über Storyboards	263
7.5.2	Das Storyboard-Projekt	263
7.5.3	Die Klasse UIStoryboard	272
7.5.4	Segues	274
7.5.5	Zugriff über den App Delegate	276
7.5.6	Quo vadis – Storyboard oder Nib-File?	277
7.6	Auto Layout	278
7.6.1	Grundlagen	278
7.6.2	Setzen und Konfigurieren von Constraints	279
7.6.3	Constraints bearbeiten und weiter anpassen	280
7.6.4	„Optimale“ Constraints automatisch setzen lassen	282
7.7	UIViewController und seine Subklassen	283
7.7.1	Grundlagen	283
7.7.2	UINavigationController	283
7.7.3	UITabBarController	288
7.7.4	UITableViewController	291
7.7.5	UICollectionViewController	297
7.7.6	UISplitViewController	298
8	Views erstellen und gestalten	301
8.1	Über Views in iOS	301
8.2	UIView – die Mutter aller Views	301
8.3	Arbeiten mit UIView	302
8.3.1	Programmatisches Erstellen einer UIView	302
8.3.2	View-Hierarchien	303
8.3.3	Weiterführendes zu UIView	307
8.4	Views erstellen mit dem Interface Builder	307
8.4.1	Grundlagen	307
8.4.2	View-Klasse mit Nib-File erstellen	308
8.4.3	Beliebiges Nib-File laden und verwenden	312
8.4.4	Nib-File nachträglich erstellen	313
8.4.5	Unterschiedliche Nib-Files für iPhone und iPad erstellen	315
8.5	Die wichtigsten Views und ihre Funktionen	316
8.5.1	Grundlagen	316
8.5.2	UILabel	316
8.5.3	UIButton	317
8.5.4	UISwitch	317
8.5.5	UISegmentedControl	317
8.5.6	UITextField	318
8.5.7	UIImageView	318

8.5.8	UIAlertView	318
8.5.9	UIActionSheet	320
8.5.10	UIPickerView	320
8.5.11	UIDatePicker	320
8.5.12	UIWebView	321
8.5.13	UIImageView	321
8.5.14	UIScrollView	321
8.5.15	UITextView	323
8.5.16	UITableView	323
8.5.17	UICollectionView	323
8.5.18	Wichtig und unerlässlich: die Dokumentation!	324
8.5.19	Views und der Interface Builder	324
8.6	Die Grundlage gut gestalteter Views	325
9	Das Model und die Datenhaltung	327
9.1	Die Logik Ihrer App	327
9.2	Benutzereinstellungen sichern und nutzen	328
9.2.1	Grundlagen	328
9.2.2	Über NSUserDefaults	328
9.2.3	Standardeinstellungen festlegen	330
9.2.4	NSUserDefaults zurücksetzen	331
9.3	Zugriff auf das Dateisystem	331
9.3.1	Das Dateisystem von iOS	331
9.3.2	NSFileManager	332
9.3.3	File-Sharing-Funktion nutzen	337
9.4	Core Data	338
9.4.1	Datenbankverwaltung mit Core Data	338
9.4.2	Wie funktioniert Core Data?	339
9.4.3	Die Klassen und Bestandteile von Core Data	339
9.4.4	Aufbau eines Standard-Core Data Stacks	341
9.4.5	Der Core Data-Editor	342
9.4.6	Erstellen eines neuen Managed-Objects	350
9.4.7	Löschen eines Managed-Objects	351
9.4.8	Laden von Managed-Objects	351
9.4.9	Was kommt als Nächstes?	352
10	Unit-Tests	353
10.1	Unit-Tests in der iOS-Entwicklung	353
10.1.1	Grundlagen	353
10.1.2	Aufbau und Funktionsweise von Tests	356
10.1.3	Aufbau einer Test-Case-Klasse	358
10.1.4	Neue Test-Case-Klasse erstellen	359
10.1.5	Ausführen von Tests	360
10.1.6	Was sollte ich eigentlich testen?	361

10.2 Performance-Tests	362
10.3 Test-Driven Development	363
11 Versionierung mit Git	365
11.1 Über Versionskontrolle	365
11.2 Basisfunktionen und -begriffe von Git	365
11.2.1 Begriffe	365
11.2.2 Funktionen	366
11.3 Source Control in Xcode	367
11.4 Version Editor und Source Control	371
12 Veröffentlichung im App Store	375
12.1 Zertifikate, Provisioning Profiles und Ihre App	375
12.1.1 Grundlagen	375
12.1.2 Certificates, Identifiers & Profiles	378
12.1.3 Erstellen von	380
12.2 Testen auf dem eigenen Endgerät	391
12.2.1 Grundlagen	391
12.2.2 Setzen des Teams	391
12.2.3 Auswahl Ihres iOS-Geräts	392
12.3 iTunes Connect und Veröffentlichung im App Store	394
12.3.1 Grundlagen	394
12.3.2 Vorbereiten der App in iTunes Connect	396
12.3.3 Upload der App in den App Store	399
12.3.4 Wie geht es weiter?	400
Index	401