Inhalt

Vorwort ..... **XIX**

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 neue 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 10
1.6.2 Apples Developer-Website 11
1.6.3 Swift.org 12
1.6.4 In eigener Sache 13

2 Grundlagen der Programmierung **15**

2.1 Grundlegendes 15
2.1.1 Swift Standard Library 15
2.1.2 print 17
2.1.3 Befehle und Semikolons 18
2.1.4 Operatoren 19
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 24
2.3 Kommentare 25

Bibliografische Informationen
http://d-nb.info/1245471260

digitalisiert durch DEUTSCHE NATIONAL BIBLIOTHEK

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 ... 30
3.2 Abfragen ... 31
3.2.1 If ... 31
3.2.2 Switch ... 36
3.2.3 Guard ... 39
3.3 Control Transfer Statements ... 41
3.3.1 Anstoßen eines neuen Schleifendurchlaufs mit continue ... 41
3.3.2 Verlassen der kompletten Schleife mit break ... 42
3.3.3 Labeled Statements ... 43

4 Typen in Swift ... 45
4.1 Integer ... 46
4.2 Fließkommazahlen ... 48
4.3 Bool ... 49
4.4 String ... 49
4.4.1 Erstellen eines Strings ... 49
4.4.2 Zusammenfügen von Strings ... 50
4.4.3 Character auslesen ... 51
4.4.4 Character mittels Index auslesen ... 52
4.4.5 Character entfernen und hinzufügen ... 53
4.4.6 Anzahl der Character zählen ... 55
4.4.7 Präfix und Suffix prüfen ... 55
4.4.8 String Interpolation ... 55
4.5 Array ... 56
4.5.1 Erstellen eines Arrays ... 57
4.5.2 Zusammenfügen von Arrays ... 58
4.5.3 Inhalte eines Arrays leeren ... 58
4.5.4 Prüfen, ob ein Array leer ist ... 59
4.5.5 Anzahl der Elemente eines Arrays zählen ... 59
4.5.6 Zugriff auf die Elemente eines Arrays ... 60
4.5.7 Neue Elemente zu einem Array hinzufügen ... 60
4.5.8 Bestehende Elemente aus einem Array entfernen ... 61
4.5.9 Bestehende Elemente eines Arrays ersetzen ... 62
4.5.10 Alle Elemente eines Arrays auslesen und durchlaufen ... 63
4.6 Set ... 64
4.6.1 Erstellen eines Sets ... 64
4.6.2 Inhalte eines bestehenden Sets leeren ... 65
4.6.3 Prüfen, ob ein Set leer ist ... 66
4.6.4 Anzahl der Elemente eines Sets zählen ... 66

4.6.5 Element zu einem Set hinzufügen ... 66
4.6.6 Element aus einem Set entfernen ... 67
4.6.7 Prüfen, ob ein bestimmtes Element in einem Set vorhanden ist ... 67
4.6.8 Alle Elemente eines Sets auslesen und durchlaufen ... 67
4.6.9 Sets miteinander vergleichen ... 68
4.6.10 Neue Sets aus bestehenden Sets erstellen ... 71
4.7 Dictionary ... 72
4.7.1 Erstellen eines Dictionaries ... 73
4.7.2 Prüfen, ob ein Dictionary leer ist ... 74
4.7.3 Anzahl der Schlüssel-Wert-Paare eines Dictionaries zählen ... 74
4.7.4 Wert zu einem Schlüssel eines Dictionaries auslesen ... 74
4.7.5 Neues Schlüssel-Wert-Paar zu Dictionary hinzufügen ... 75
4.7.6 Bestehendes Schlüssel-Wert-Paar aus Dictionary entfernen ... 76
4.7.7 Bestehendes Schlüssel-Wert-Paar aus Dictionary verändern ... 76
4.7.8 Alle Schlüssel-Wert-Paare eines Dictionaries auslesen und durchlaufen ... 77
4.8 Tuple ... 78
4.8.1 Zugriff auf die einzelnen Elemente eines Tuples ... 79
4.8.2 Tuple und switch ... 80
4.9 Optional ... 83
4.9.1 Deklaration eines Optionals ... 84
4.9.2 Zugriff auf den Wert eines Optionals ... 84
4.9.3 Optional Binding ... 86
4.9.4 Implicitly Unwrapped Optional ... 88
4.9.5 Optional Chaining ... 89
4.9.6 Optional Chaining über mehrere Eigenschaften und Funktionen ... 93
4.10 Any und AnyObject ... 97
4.11 Type Alias ... 98
4.12 Value Type versus Reference Type ... 98
4.12.1 Reference Types auf Gleichheit prüfen ... 100

5 Funktionen ... 103
5.1 Funktionen mit Parametern ... 104
5.1.1 Argument Labels und Parameter Names ... 105
5.1.2 Default Value für Parameter ... 108
5.1.3 Variadic Parameter ... 109
5.1.4 In-Out-Parameter ... 110
5.2 Funktionen mit Rückgabewert ... 111
5.3 Function Types ... 113
5.3.1 Funktionen als Variablen und Konstanten ... 115
5.4 Verschachtelte Funktionen ... 116
5.5 Closures ... 117

5.5.1 Closures als Parameter von Funktionen ... 118
5.5.2 Trailing Closures ... 121
5.5.3 Autoclosures ... 122

6 Enumerations, Structures und Classes ... 125
6.1 Enumerations ... 125
6.1.1 Enumerations und switch ... 128
6.1.2 Associated Values ... 129
6.1.3 Raw Values ... 131
6.2 Structures ... 134
6.2.1 Erstellen von Structures und Instanzen ... 134
6.2.2 Eigenschaften und Funktionen ... 135
6.3 Classes ... 140
6.3.1 Erstellen von Klassen und Instanzen ... 141
6.3.2 Eigenschaften und Funktionen ... 141
6.4 Enumeration vs. Structure vs. Class ... 143
6.4.1 Gemeinsamkeiten und Unterschiede ... 143
6.4.2 Wann nimmt man was? ... 144
6.5 self ... 146

7 Eigenschaften und Funktionen von Typen ... 149
7.1 Properties ... 149
7.1.1 Stored Property ... 150
7.1.2 Lazy Stored Property ... 152
7.1.3 Computed Property ... 155
7.1.4 Read-Only Computed Property ... 159
7.1.5 Property Observer ... 160
7.1.6 Property Wrapper ... 163
7.1.7 Type Property ... 170
7.2 Globale und lokale Variablen ... 172
7.3 Methoden ... 175
7.3.1 Instance Methods ... 175
7.3.2 Type Methods ... 178
7.4 Subscripts ... 179

8 Initialisierung ... 185
8.1 Aufgabe der Initialisierung ... 186
8.2 Erstellen eigener Initializer ... 187
8.3 Initializer Delegation ... 193
8.3.1 Initializer Delegation bei Value Types ... 193
8.3.2 Initializer Delegation bei Reference Types ... 194
8.4 Failable Initializer ... 197

8.5 Required Initializer ... 200
8.6 Deinitialisierung ... 200

9 Vererbung ... 203
9.1 Überschreiben von Eigenschaften und Funktionen einer Klasse ... 206
9.2 Überschreiben von Eigenschaften und Funktionen einer Klasse verhindern ... 209
9.3 Zugriff auf die Superklasse ... 209
9.4 Initialisierung und Vererbung ... 210
9.4.1 Zwei-Phasen-Initialisierung ... 211
9.4.2 Überschreiben von Initializern ... 218
9.4.3 Vererbung von Initializern ... 220
9.4.4 Required Initializer ... 221

10 Speicherverwaltung mit ARC ... 223
10.1 Strong Reference Cycles ... 226
10.1.1 Weak References ... 228
10.1.2 Unowned References ... 232
10.1.3 Weak Reference vs. Unowned Reference ... 234

11 Weiterführende Sprachmerkmale von Swift ... 235
11.1 Nested Types ... 235
11.2 Extensions ... 237
11.2.1 Computed Properties ... 237
11.2.2 Methoden ... 238
11.2.3 Initializer ... 239
11.2.4 Subscripts ... 241
11.2.5 Nested Types ... 242
11.3 Protokolle ... 242
11.3.1 Deklaration von Eigenschaften und Funktionen ... 244
11.3.2 Der Typ eines Protokolls ... 254
11.3.3 Protokolle und Extensions ... 256
11.3.4 Vererbung in Protokollen ... 260
11.3.5 Class-only-Protokolle ... 261
11.3.6 Optionale Eigenschaften und Funktionen ... 263
11.3.7 Protocol Composition ... 265
11.3.8 Delegation ... 267
11.3.9 Übersicht diverser vorhandener Protokolle ... 269
11.4 Key-Path ... 271

12 Type Checking und Type Casting ... 275
12.1 Type Checking mit „is“ ... 278
12.2 Type Casting mit „as“ ... 279

13 Error Handling ... 281
13.1 Deklaration und Feuern eines Fehlers ... 281
13.2 Reaktion auf einen Fehler ... 285
13.2.1 Mögliche Fehler mittels do-catch auswerten ... 285
13.2.2 Mögliche Fehler in Optionals umwandeln ... 289
13.2.3 Mögliche Fehler weitergeben ... 289
13.2.4 Mögliche Fehler ignorieren ... 291

14 Generics ... 293
14.1 Generic Functions ... 294
14.2 Generic Types ... 297
14.3 Type Constraints ... 300
14.4 Associated Types ... 300

15 Nebenläufigkeit ... 305
15.1 Asynchronen Code schreiben und aufrufen ... 305
15.2 Mehrere asynchrone Funktionen parallel ausführen ... 308
15.3 Actors ... 310

16 Dateien und Interfaces ... 313
16.1 Modules und Source Files ... 313
16.2 Access Control ... 314
16.2.1 Access Level ... 314
16.2.2 Explizite und implizite Zuweisung eines Access Levels ... 318
16.2.3 Besonderheiten ... 319

Teil II: Xcode ... 325

17 Grundlagen, Aufbau und Einstellungen von Xcode ... 327
17.1 Über Xcode ... 328
17.2 Arbeiten mit Xcode ... 329
17.2.1 Dateien und Formate eines Xcode-Projekts ... 329
17.2.2 Umgang mit Dateien und Ordnern ... 334
17.3 Der Aufbau von Xcode ... 339
17.3.1 Toolbar ... 339
17.3.2 Navigator ... 342
17.3.3 Editor ... 346
17.3.4 Inspectors ... 350
17.3.5 Debug Area ... 353
17.4 Einstellungen ... 354
17.4.1 General ... 354
17.4.2 Accounts ... 355

17.4.3 Behaviors ... 356
17.4.4 Navigation ... 356
17.4.5 Themes ... 357
17.4.6 Text Editing ... 358
17.4.7 Key Bindings ... 358
17.4.8 Source Control ... 359
17.4.9 Components ... 360
17.4.10 Locations ... 360
17.4.11 Server & Bots ... 361
17.5 Projekteinstellungen ... 362
17.5.1 Einstellungen am Projekt ... 363
17.5.2 Einstellungen am Target ... 365
17.5.3 Einstellungen am Scheme ... 373

18 Dokumentation, Devices und Organizer ... 379
18.1 Dokumentation ... 379
18.1.1 Aufbau und Funktionsweise ... 380
18.1.2 Direktzugriff im Editor ... 383
18.2 Devices und Simulatoren ... 385
18.2.1 Simulatoren ... 387
18.2.2 Devices ... 389
18.3 Organizer ... 391

19 Debugging und Refactoring ... 393
19.1 Debugging ... 393
19.1.1 Konsolenausgaben ... 394
19.1.2 Arbeiten mit Breakpoints ... 395
19.1.3 Debug Navigator ... 400
19.2 Refactoring ... 402
19.3 Instruments ... 404

20 Tipps und Tricks für das effiziente Arbeiten mit Xcode ... 409
20.1 Code Snippets ... 409
20.2 Open Quickly ... 411
20.3 Related Items ... 412
20.4 Navigation über die Jump Bar ... 413
20.5 MARK, TODO und FIXME ... 413
20.6 Shortcuts für den Navigator ... 415
20.7 Clean Build ... 415
20.8 Playgrounds ... 415

Teil III: App-Entwicklung **419**

21 Grundlagen der App-Entwicklung **421**

21.1 Die Basis: SwiftUI 421
21.2 Bestandteile einer App 423
21.2.1 Umsetzung der Daten 424
21.2.2 Umsetzung der Ansichten 424
21.2.3 Weitere Frameworks 424
21.3 Die Syntax von SwiftUI 424
21.4 Aufbau einer App 425
21.5 Das View-Protokoll 426
21.6 Aktualisierung von Views mittels Status 427
21.7 Grundlagen des Status 429
21.8 Anpassung von Views mittels Modifier 432
21.9 Gruppierung von Views mittels Containern 435
21.10 Praxis: Unsere erste App 436
21.10.1 Bestandteile des neuen Projekts 442
21.10.2 Änderung des Textes 444
21.10.3 Einsatz der Preview 445

22 Views in SwiftUI **449**

22.1 Textdarstellung und -bearbeitung 449
22.1.1 Text 449
22.1.2 TextField 459
22.1.3 SecureField 462
22.1.4 TextEditor 463
22.2 Bilder 466
22.2.1 Image-Instanz erstellen 468
22.2.2 Größe einer Image-Instanz ändern 470
22.3 Schaltflächen 472
22.3.1 Button 473
22.3.2 EditButton 476
22.3.3 PasteButton 477
22.4 Werteauswahl 479
22.4.1 Toggle 479
22.4.2 Slider 484
22.4.3 Stepper 490
22.4.4 Picker 495
22.4.5 DatePicker 499
22.4.6 ColorPicker 503
22.5 Werteindikatoren 506
22.5.1 Label 506
22.5.2 ProgressView 509
22.5.3 Gauge 512

23 View-Layout ... 517
23.1 Stacks ... 517
23.1.1 HStack ... 517
23.1.2 VStack ... 521
23.1.3 ZStack ... 525
23.1.4 Stacks verschachteln ... 526
23.1.5 Lazy Stacks ... 527
23.2 Listen ... 531
23.2.1 List ... 531
23.2.2 ForEach ... 552
23.3 Grids ... 563
23.4 Container-Views ... 578
23.4.1 Form ... 579
23.4.2 Section ... 581
23.4.3 Group ... 584
23.4.4 GroupBox ... 589
23.5 Weitere Views ... 592
23.5.1 ScrollView ... 592
23.5.2 OutlineGroup ... 596
23.5.3 DisclosureGroup ... 601
23.5.4 Spacer ... 606
23.5.5 Divider ... 609

24 Navigation ... 611
24.1 NavigationView und NavigationLink ... 611
24.1.1 Festlegen einer Standard-View für die Detailansicht ... 618
24.1.2 NavigationView-Style anpassen ... 620
24.1.3 NavigationView-Titel setzen ... 623
24.1.4 Navigation-Bar ausblenden ... 626
24.1.5 Eigene View zur Darstellung eines NavigationLink nutzen ... 628
24.1.6 NavigationLink programmatisch ausführen ... 630
24.1.7 Verhalten eines NavigationLink unter iPadOS anpassen ... 636
24.2 TabView ... 640
24.3 HSplitView und VSplitView ... 648
24.4 Funktionen zur Präsentation von Views ... 648
24.4.1 Sheet einblenden ... 649
24.4.2 View über gesamtes Fenster legen ... 655
24.4.3 Popover einblenden ... 660

25 Weitere View-Konfigurationen ... 667
25.1 Toolbar ... 667
25.2 Alerts ... 675

25.3 Confirmation Dialog ... 678
25.4 Farben ... 680
25.5 View-Events ... 681

26 Status ... 683
26.1 Property ... 685
26.2 State ... 687
26.3 Binding ... 688
26.4 ObservedObject ... 697
26.4.1 Datenmodell vorbereiten ... 698
26.4.2 Datenmodell in View einbinden ... 699
26.4.3 Auf Änderungen reagieren ... 702
26.5 StateObject ... 706
26.6 EnvironmentObject ... 707
26.7 Environment ... 713
26.8 SceneStorage ... 717
26.9 AppStorage ... 719
26.10 Source of Truth vs. Derived Value ... 721
26.11 Best Practices ... 724

27 Datenhaltung ... 729
27.1 UserDefaults ... 729
27.1.1 UserDefaults und SwiftUI ... 731
27.2 Core Data ... 732
27.2.1 Grundlegende Funktionsweise von Core Data ... 732
27.2.2 Grundlegende Elemente beim Einsatz von Core Data ... 733
27.2.3 Einen Core Data Stack erstellen ... 735
27.2.4 Ein Managed Object Model erstellen ... 737
27.2.5 Grundlegende Core-Data-Operationen ... 747
27.2.6 Core Data mit SwiftUI ... 749

28 Weitere Projektkonfigurationen ... 757
28.1 Cross-Platform-Entwicklung ... 757
28.1.1 Neue Targets hinzufügen ... 757
28.1.2 Target-Zuweisung ... 760
28.1.3 Plattform im Code prüfen ... 761
28.2 Mehrsprachigkeit ... 762
28.2.1 Grundlagen ... 762
28.2.2 Übersetzungen mit SwiftUI ... 767
28.2.3 Verschiedene Sprachen einer App testen ... 768
28.3 Asset Catalogs ... 769

29 Preview und Library ... 773
29.1 Preview ... 773
29.1.1 Funktionsweise der Preview ... 775
29.1.2 Konfiguration der Preview ... 777
29.1.3 Preview ausführen ... 778
29.2 Library ... 779
29.3 Attributes Inspector ... 781

Teil IV: Source Control und Testing ... 783

30 Source Control ... 785
30.1 Basisfunktionen und -begriffe der Source Control ... 785
30.2 Source Control in Xcode ... 787
30.2.1 Bestehendes Projekt klonen ... 788
30.2.2 Lokale Änderungen committen ... 790
30.2.3 Lokale Änderungen verwerfen ... 791
30.2.4 Pull und Push ... 792
30.2.5 Aktuelle Branches vom Repository laden ... 793
30.2.6 Git-Repository mit neuem Xcode-Projekt erzeugen ... 793
30.2.7 Optische Source-Control-Hervorhebungen im Editor ... 794
30.2.8 Zugriff auf GitHub, GitLab und Bitbucket ... 795
30.3 Source Control Navigator ... 796
30.4 Code Review-Mode ... 797

31 Testing ... 801
31.1 Unit-Tests ... 801
31.1.1 Aufbau und Funktionsweise von Unit-Tests ... 805
31.1.2 Aufbau einer Test-Case-Klasse ... 808
31.1.3 Neue Test-Case-Klasse erstellen ... 810
31.1.4 Ausführen von Unit-Tests ... 811
31.1.5 Was sollte ich eigentlich testen? ... 813
31.2 Performancetests ... 814
31.3 UI-Tests ... 815
31.3.1 Klassen für UI-Tests ... 817
31.3.2 Aufbau von UI-Test-Klassen ... 819
31.3.3 Automatisches Erstellen von UI-Tests ... 819
31.3.4 Einsatz von UI-Tests ... 820

Teil V: Veröffentlichung von Apps **821**

32 Veröffentlichung im App Store **823**

32.1 Das Apple Developer Portal 824
32.1.1 Zertifikate, App IDs und Provisioning Profiles 827
32.1.2 Code Signing 842
32.2 App Store Connect 845
32.2.1 Apps für den App Store vorbereiten und verwalten 846
32.2.2 Apps erstellen, hochladen und einreichen 850
32.3 App Store Review Guidelines 852

33 Das Business Model für Ihre App **855**

33.1 Geschäftsmodelle 855
33.1.1 Free Model 855
33.1.2 Freemium Model 856
33.1.3 Subscription Model 856
33.1.4 Paid Model 857
33.1.5 Paymium Model 857
33.2 App Bundles 858
33.3 Veröffentlichung außerhalb des App Store 859
33.3.1 Das Apple Developer Enterprise Program 860

34 TestFlight **863**

34.1 TestFlight in App Store Connect 863
34.2 TestFlight im App Store 865

Index **867**