Inhalt

TEIL II App-Programmierung

Vorwort

Apples größte Innovation des Jahres 2014 war aus meiner Sicht weder die Vorstellung der Apple Watch noch die Auslieferung des Bestsellers iPhone 6. Apple hat sich neben den Arbeiten an diesen Produkten einer anderen Baustelle zugewandt und als Reaktion auf die vielen Mängel, die die rund 20 Jahre alte Programmiersprache Objective-C aufweist, eine vollkommen neue Programmiersprache entwickelt: Swift!

In ersten Kommentaren konnten selbst Apple-Fans ihre Skepsis nicht verbergen: Brauchen wir wirklich eine neue Programmiersprache? Doch je mehr Details Apple auf der World Wide Developers Conference (WWDC 2014) verriet, desto größer wurde die Begeisterung der teilnehmenden Entwickler und der Fachpresse.

Warum Swift?

Swift ist für Apple ein Befreiungsschlag: Objective-C dient dem Apple-Universum seit vielen Jahren als Fundament. Das ändert aber nichts daran, dass Objective-C eine Programmiersprache aus den 1980er-Jahren ist, die in keinerlei Hinsicht mit modernen Programmiersprachen mithalten kann.

Swift ist dagegen ein sauberer Neuanfang. Bei der Vorstellung wurde Swift auch *Objective-C without the C* genannt. Natürlich ist Swift von Objective-C beeinflusst – schließlich muss Swift kompatibel zu den unzähligen Apple-Bibliotheken sein. Swift realisiert viele neue Ideen, greift aber auch Konzepte von C#, Haskell, Java, Python und anderen Programmiersprachen auf. Daraus ergeben sich mehrere Vorteile:

- Swift zählt zu den modernsten Programmiersprachen, die es momentan gibt.
- Code lässt sich in Swift syntaktisch eleganter formulieren als in Objective-C.
- Der resultierende Code ist besser lesbar und wartbar.
- Swift ist für Programmierer, die schon Erfahrung mit anderen modernen Sprachen gesammelt haben, wesentlich leichter zu erlernen als Objective-C. Vorhandenes Know-how lässt sich einfacher auf Swift als auf Objective-C übertragen.
- Swift ist ein Open-Source-Produkt und steht auch für Linux zur Verfügung. Der Entwicklungsprozess erfolgt offen und transparent.

Swift ist in den letzten Jahren kometenhaft in die Top-Listen der populärsten Programmiersprachen aufgestiegen. Im TIOBE-Index war Swift zuletzt auf Platz 11 vertreten.

Neu in Swift 4

Langjährige Swift-Entwickler haben Swift 3 in schlechter Erinnerung: Unzählige inkompatible Neuerungen erforderten umfassende Änderungen an vorhandenem Code. Wesentlich besser sieht es bei Swift 4 aus: Ein Großteil der Neuerungen sind Erweiterungen, die keine Inkompatibilitäten verursachen. Und da, wo doch Code-Änderungen erforderlich sind, kümmert sich in der Regel der Code-Konverter von Xcode um die Anpassungen. Kurzum: Die Anpassung von Swift-3-Code an Swift 4 sollte keine großen Probleme verursachen. Ärger lauert allerdings an anderer Stelle: Veränderte Frameworks und APIs sowie Methoden und Klassen, die plötzlich als *deprecated* gelten, zwingen in manchen Apps zu größeren Umbauten. Das wiederum ist aber nicht die Schuld von Swift.

Was sind nun die wichtigsten Verbesserungen in Swift 4?

- **Zeichenketten:** Der Umgang mit Zeichenketten ist ein wenig komfortabler geworden: Zum einen lassen sich jetzt sehr elegant mehrzeilige Zeichenketten formulieren, zum anderen ist die ständige Nennung der Eigenschaft überflüssig, um auf die Sequenz der Zeichen einer Zeichenkette zuzugreifen. Die neue `Substring`-Struktur macht die Verarbeitung von Teilzeichenketten effizienter. Weitere Optimierungen wurden im Hintergrund vorgenommen.
- **Bereiche (Ranges):** Bereiche können nun zur einen oder anderen Seite offen formuliert werden, also in der Form `ar[3...]` (alle Array-Elemente ab dem vierten) oder `s[..<endpos]` (die Teilzeichenkette vom Beginn von `s` bis zur Position `endpos`).
- **Generics:** Die Syntax zur Formulierung generischer Ausdrücke wurde an einigen Punkten erweitert und gibt nun mehr Spielraum als bisher.
- **JSON-Support:** Die Swift-Standardbibliothek wurde um das Protokoll `Codable` sowie um diverse verwandte Protokolle und Klassen erweitert. Sie machen es möglich, eigene Datentypen mit minimalem Aufwand im JSON-Format zu speichern bzw. aus JSON-Dateien wieder einzulesen.
- **Xcode:** Der Code-Editor von Xcode wurde komplett neu implementiert. Das merkt man an der höheren Geschwindigkeit, an diversen Instabilitäten, vor allem aber daran, dass es nun endlich Refactoring-Kommandos gibt. Hurra!

Keine Angst vor Swift 5!

Für mich als Autor ist es jedes Jahr ein wenig befremdlich: Ich arbeite mit voller Energie an der Fertigstellung meines Buchs zu Swift 4, da diskutieren die Entwickler bereits über Swift 5! Diese Version soll noch mehr Optimierungen beim Umgang mit Zeichenketten bringen, weitere Generics-Features sowie ein Fundament für asynchrone Programmierkonstrukte. (Deren vollständige Implementierung ist allerdings erst für Swift 6 geplant.)

Generell ist geplant, inkompatible Änderungen auf ein absolutes Minimum zu beschränken. Swift wird also erweitert, aber kaum mehr geändert. Insofern müssen Sie sich nicht vor Swift 5

fürchten: Soweit es wirklich Änderungen gibt, wird der in Xcode integrierte Code-Konverter in bewährter Manier einen Großteil der Änderungen automatisch durchführen.

Außerdem hat sich bei meiner Arbeit an den drei Auflagen dieses Buchs eines herauskristallisiert: Bei der App-Programmierung kostet nicht der Umgang mit Swift an sich Zeit, sondern die Suche nach den geeigneten Klassen, Methoden und Programmiertechniken.

Was bietet dieses Buch?

Dieses Buch vermittelt einen kompakten Einstieg in die Programmiersprache Swift in der Version 4 (Xcode 9). Das Buch ist in vier Teile gegliedert:

- **Teil I** führt in die Grundlagen von Swift ein. Hier lernen Sie alle wichtigen Sprachdetails kennen. Die Themenpalette reicht vom Umgang mit Variablen und elementaren Datentypen bis hin zur Syntax der objekt- und protokollorientierten Programmierung.
- **Teil II** ist eine Einführung in die Entwicklung von Apps für iOS, macOS und tvOS. Hier erkläre ich Ihnen beispielsweise, wie der Storyboard-Editor funktioniert, wie Sie Ihre Oberfläche mit eigenem Swift-Code verbinden, eigene `ViewController`-Klassen entwickeln, Apps mit mehreren Dialogen/Views organisieren etc.
- **Teil III** fasst wichtige Programmiertechniken in Bausteinform zusammen. In Kurzanleitungen zeige ich Ihnen unter anderem, wie Sie auf Dateien zugreifen, XML-Dokumente auswerten, Webseiten anzeigen, Steuerelemente mit eigener Grafik gestalten, Listen und Tabellen in Apps darstellen, geografische Daten auswerten und Spiele mit SpriteKit programmieren. Sobald Ihre App zufriedenstellend funktioniert, lernen Sie, wie Sie sie tauglich für den App Store machen und dort einreichen.
- **Teil IV** zeigt anhand konkreter Beispielprojekte die Praxis der App-Programmierung. Die Apps decken eine ganze Palette von Themen ab: vom praktischen Währungsumrechner über den Icon-Resizer bis hin zu mehreren Spielen.

Neu in dieser Auflage sind nicht nur Swift-4-Features. Weitere inhaltliche Schwerpunkte sind Drag & Drop für iOS, die iCloud-Programmierung sowie Server-side Swift.

Selbstverständlich können Sie alle Beispieldateien und -projekte dieses Buchs herunterladen. Einen Download-Link finden Sie hier:

www.rheinwerk-verlag.de/4494

Um von diesem Buch maximal zu profitieren, benötigen Sie weder Vorkenntnisse in Xcode noch in der App-Entwicklung. Ich setze aber voraus, dass Sie bereits Erfahrungen mit einer beliebigen Programmiersprache gesammelt haben. Ich erkläre Ihnen in diesem Buch also, wie Sie in Swift mit Variablen umgehen, Schleifen programmieren und Klassen entwickeln, aber nicht, was Variablen sind, wozu Schleifen dienen und warum Klassen das Fundament der objektorientierten Programmierung sind. So kann ich Swift kompakt und ohne viel Overhead beschreiben und den Schwerpunkt auf die konkrete Anwendung legen.

Leseanleitung

1300 Seiten – das kann schon abschrecken! Dazu besteht aber kein Grund. Ich habe mich beim Schreiben dieses Buchs bemüht, den Inhalt auf möglichst eigenständige Kapitel zu verteilen, aus denen Sie sich wie aus einem Baukasten bedienen können.

Wenn Swift für Sie vollständig neu ist, dann ist die Lektüre der ersten Kapitel aus Teil I natürlich unumgänglich. Besonders wichtig ist, dass Sie die Swift-spezifischen Eigenheiten beim Umgang mit elementaren Datentypen und Aufzählungen (Arrays, Dictionaries etc.) kennenlernen und das Konzept von Optionals verstehen. Interessanterweise hat sich herausgestellt, dass Sie für die Entwicklung erster Apps nicht unbedingt alle Feinheiten im Zusammenhang mit Vererbung, Protokollen etc. beherrschen müssen. Die Basics reichen zumeist.

Teil II richtet sich speziell an Programmierer, die erstmalig Apps für iOS, macOS oder tvOS entwickeln. Wenn Sie bisher Objective-C zur App-Programmierung verwendet haben, werden Sie in Teil II auf viel bekanntes Wissen stoßen.

Bei Teil III habe ich versucht, oft benötigte Programmiertechniken möglichst losgelöst von der Zielplattform zu beschreiben. Beispielsweise erfolgt der Umgang mit Dateien unter iOS ganz ähnlich wie unter macOS. Aus den Kapiteln in Teil III können Sie sich also bedienen, wie Sie es gerade brauchen.

Viele Detailprobleme treten erst dann auf, wenn man den Schritt von kleinen Beispielen hin zu »richtigen« Apps macht. Deswegen stellt Teil IV eine Reihe vollständiger Projekte vor. Auch wenn es unwahrscheinlich ist, dass Sie genau so eine App programmieren möchten wie eines der Beispiele aus Teil IV, so werden Sie in diesen Kapiteln vermutlich doch inhaltlich verwandte Anleitungen und Arbeitstechniken mit einem hohen Praxisbezug finden. Probieren Sie die Apps einfach einmal aus, und blättern Sie dann durch die entsprechenden Kapitel – Sie werden sicher über Details stolpern, die sich später als hilfreich herausstellen werden.

Viel Spaß bei der App-Entwicklung!

Eine neue Programmiersprache zu erlernen ist immer eine Herausforderung. Noch schwieriger ist es, einen Überblick über die schier unüberschaubare Fülle von Bibliotheken zu gewinnen, die Sie zur App-Entwicklung brauchen. Dieses Buch soll Ihnen bei beiden Aspekten helfen und Ihnen ein solides Fundament vermitteln.

Wenn Sie in die App-Entwicklung mit Swift einsteigen, haben Sie das Privileg, mit einer der modernsten aktuell verfügbaren Programmiersprachen zu arbeiten. Sobald Sie die ersten Schritte einmal erfolgreich absolviert haben, wird die Faszination für diese Sprache auch Sie erfassen. Bei Ihrer Reise durch die neue Welt von Swift wünsche ich Ihnen viel Spaß und Erfolg!

Michael Kofler (*https://kofler.info*)

PS: Vielen Dank an Alfred Schilken für sein Feedback zum Manuskript!