Swift
im Detail

thomas **SILLMANN**

HANSER

Sillmann

Swift im Detail

Thomas Sillmann

Swift im Detail

HANSER

Der Autor:
Thomas Sillmann, Aschaffenburg
www.thomassillmann.de

Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

Lektorat: Sieglinde Schärl
Copy editing: Sandra Gottmann, Münster-Nienberge
Herstellung: Irene Weilhart
Umschlagdesign: Marc Müller-Bremer, München, www.rebranding.de
Umschlagrealisation: Stephan Rönigk
Gesamtherstellung: Kösel, Krugzell
Printed in Germany

Print-ISBN: 978-3-446-44294-8
E-Book-ISBN: 978-3-446-44423-2

Für Ela

– auf das, was war, und das, was sein wird.

Inhalt

1 Apples neue Programmiersprache: Swift

1.1 Willkommen bei Swift!

Das war schon eine ziemliche Überraschung im Juni 2014 zu Beginn von Apples alljährlicher Entwicklerkonferenz WWDC. Ich als Entwickler habe mit vielem gerechnet, allen voran mit der Vorstellung der neuen Betriebssystemversionen von OS X und iOS. Auch eine iWatch (inzwischen als Apple Watch bekannt) hätte ich mir damals bereits vorstellen können. Doch dass Apple dann nicht nur Unmengen neuer APIs und Frameworks für Entwickler aus dem Hut zaubert, sondern gleich noch eine komplett neue Programmiersprache vorstellt, hat wahrlich meine kühnsten Vorstellungen übertroffen und meine Kinnlade während der Präsentation weit herabsinken lassen. Immerhin bin ich damit nicht alleine, denn Swift war eine spannende und gänzlich unerwartete Vorstellung auf der WWDC 2014. Die Begeisterung und Faszination für diese Sprache ist seitdem ungebrochen und es gibt kaum einen iOS- und OS X-Entwickler, der sich nicht ausgiebig mit Apples neuer Programmiersprache auseinandersetzt. Nicht nur, dass Swift einfacher anzuwenden sein soll als das bisher von Apple präferierte Objective-C, nein, auch sollen zukünftig mit Swift entwickelte Apps bis zu 30 % schneller sein als ihre Objective-C-Pendants. Allein das ist Grund genug, dass sich nicht nur Einsteiger und Apple-Neulinge mit Swift intensiv auseinandersetzen.

1.2 Warum Swift?

Laut Apple begann die Entwicklung der Programmiersprache Swift im Jahr 2010. Hauptgrund für die Entwicklung von Swift dürfte unter anderem gewesen sein, dass Objective-C als bisherige Hauptsprache zur Programmierung für iOS und OS X den ein oder anderen Entwickler abschreckte. Nicht zuletzt aufgrund seiner Syntax war und ist Objective-C – gerade für Umsteiger von anderen Programmiersprachen – anfangs ein wenig befremdlich und schwer zu verstehen und zu erlernen. Eine Programmiersprache wie Java, die beispielsweise für die Entwicklung von Android-Apps genutzt wird und wesentlich weiter verbreitet ist, lockt da potenzielle App-Entwickler schon eher. Dieses Umstands war sich auch Apple bewusst, und so hat Swift auch einen klaren Grundsatz: Einfach, schnell, spaßig. Mit Swift

sollen Entwickler schnell und einfach eigene Anwendungen für iOS und OS X schreiben können und wahrlich Spaß an der Entwicklung haben.

Zu diesem Zweck hat Apple für Swift die Vorteile und Merkmale aus verschiedensten Programmiersprachen als Basis genommen und in Swift einfließen lassen. Neben Objective-C dienen so beispielsweise auch Ruby und Python als Vorlage für Swift.

Auch wenn selbstredend aktuell Objective-C bei den meisten App-Entwicklern weiter verbreitet ist, so ändert das aber nichts daran, dass Swift langfristig die bessere Alternative zu Objective-C sein dürfte. Allein die Tatsache, dass mit Swift entwickelte Apps bis zu 30 % schneller ausgeführt werden als in Objective-C geschriebene Anwendungen, ist eine deutliche Ansage. Auch macht die Einfachheit von Swift – verbunden mit allen Möglichkeiten und API-Zugriffen wie mit Objective-C auch – die Sprache ebenso für alteingesessene Entwickler attraktiv. Und ob Apple früher oder später nicht doch irgendwann den Objective-C-Hahn zudreht und nur noch Swift unterstützt, bleibt abzuwarten.

Sollten Sie jetzt ganz frisch mit der iOS- und/oder OS X-Entwicklung beginnen, kann ich Ihnen nur wärmstens empfehlen, sich voll und ganz auf Swift zu konzentrieren. Sie erlernen dann eine moderne Sprache mit moderner Architektur, die in den kommenden Jahren noch weitaus mehr Popularität und Anhänger gewinnen wird.

Und wenn Sie bereits mehrere Jahre mit Objective-C entwickeln (so wie ich auch), dann lassen Sie sich nichtsdestotrotz auf Swift ein. Starten Sie entweder ein neues Projekt mit Swift oder verwenden Sie Swift zur Erweiterung Ihrer bestehenden Anwendungen (denn Objective-C und Swift können problemlos gemischt innerhalb eines Projekts verwendet werden, aber dazu gleich mehr). Langfristig wird Swift *die* Sprache zur Entwicklung von iOS- und OS X-Apps sein, und umso früher Sie Ihre Expertise in diesem Bereich ausbauen und Erfahrungen sammeln, desto besser sind Sie auf alle Neuerungen und Erweiterungen der Sprache in Zukunft vorbereitet.

1.3 Swift und Objective-C

Wie bereits geschrieben, ist Swift zum aktuellen Zeitpunkt kein Ersatz, sondern eine neue Programmiersprache neben Objective-C zur Entwicklung von iOS- und OS X-Apps. Damit Sie sich aber nicht ausschließlich für eine der beiden entscheiden müssen, haben Sie die Möglichkeit, Swift- und Objective-C-Code parallel in einem Projekt zu verwenden. Zwar können Sie nicht innerhalb einer Quellcode-Datei Swift- und Objective-C-Code mischen, können aber in einem Projekt sowohl Swift- als auch Objective-C-Klassen verwenden, die dann jeweils Code der entsprechenden Programmiersprache enthalten (siehe Bild 1.1). Das erlaubt Ihnen, selbst bereits bestehende und in Objective-C geschriebene Anwendungen um neue Swift-Klassen und -Funktionen zu erweitern.