

# Vorwort

Liebe Leserin, lieber Leser,

ich entwickle seit inzwischen über zehn Jahren Apps für die verschiedenen Plattformen von Apple. Die Einführung des iPhone und des App Store hat meinen Werdegang maßgeblich beeinflusst und sorgte dafür, dass ich mich heute voll und ganz dem Apple-Kosmos verschrieben habe.

In all diesen Jahren gab es viele kleine Evolutionen, die uns App-Entwicklern das Leben erleichterten. Die Einführung von Automatic Reference Counting vereinfachte die Speicherverwaltung deutlich. Storyboards öffneten ganz neue Wege, App-Strukturen umzusetzen und Views zu gestalten. Auto Layout verbesserte die Möglichkeiten, Views für verschiedene Bildschirmgrößen zu optimieren.

Daneben gab es auch einige wenige *große* Revolutionen. Eine davon war die Einführung der Programmiersprache Swift. Eine andere zeichnet sich erst seit jüngster Zeit ab. Die Rede ist von *SwiftUI*.

Mit SwiftUI ändert sich maßgeblich, wie Views für die verschiedenen Plattformen von Apple umgesetzt werden. Es gibt keine View-Controller mehr, nur Views. Die basieren auf Structures, nicht auf Klassen. Ihre Erstellung erfolgt deklarativ, nicht imperativ. Und ein Status bestimmt, welches Verhalten sie an den Tag legen und unter welchen Bedingungen sie sich aktualisieren.

Die Arbeit mit SwiftUI ist so gänzlich anders als das, was man all die letzten Jahre mit AppKit, UIKit und WatchKit gewohnt ist. Gleichzeitig zeichnet sich jetzt bereits ab, wie mächtig dieses neue UI-Framework von Apple ist. Noch nie war es leichter, ansprechende Nutzeroberflächen zu erstellen. Und noch nie brauchte es dafür so wenige Zeilen Code wie mit SwiftUI.

Dazu kommt, dass SwiftUI auf allen Apple-Plattformen zur Verfügung steht. Hat man die grundlegende Funktionsweise demnach einmal verinnerlicht, ist man imstande, Views für macOS, iOS (und iPadOS), watchOS sowie tvOS zu erstellen. SwiftUI stellt ein gemeinsames Toolset dar, das sich im gesamten Apple-Kosmos nutzen lässt.

Seit der erstmaligen Vorstellung von SwiftUI auf der WWDC 2019 bin ich begeistert von diesem Framework. Wie mächtig es ist, wird mir jedes Mal bewusst, wenn ich in Projekten auf die „alten“ Techniken zur Erstellung von Nutzeroberflächen mittels Storyboards und View-Controllern zurückgreife. Im Vergleich ist die Arbeit mit SwiftUI um so vieles komfortabler.

SwiftUI stellt die Zukunft der UI-Erstellung für Apple-Plattformen dar, und mit diesem Buch möchte ich Ihnen einen passenden Einstieg zur Verfügung stellen. In den folgenden Kapiteln erfahren Sie, wie SwiftUI funktioniert und welche Views Ihnen zur Verfügung stehen. Auch gehe ich im Detail auf den Status ein und wie er sich auf die Aktualisierung von Ansichten auswirkt. Ebenso kommt die Integration von SwiftUI in bestehende Projekte auf Basis von Storyboards nicht zu kurz.

Zusätzlich erhalten Sie zusammen mit diesem Buch noch Zugriff auf einen ganz besonderen Service: Dank Update inside kommen Sie in den Genuss von Zusatzkapiteln, die nach und nach veröffentlicht werden. Neben weiteren Themen, die es aus Platzgründen nicht mehr in dieses Buch geschafft haben, werden Sie so auch über kommende SwiftUI-Updates informiert. Der Update-Service läuft bis Oktober 2022. Sie werden persönlich von uns benachrichtigt, wenn neue Updates zum Download zur Verfügung stehen. Registrieren Sie sich dazu einfach unter [www.hanser-fachbuch.de/swiftui-update](http://www.hanser-fachbuch.de/swiftui-update) mit dem Passwort von der zweiten Seite dieses Buches.

Nun bleibt mir nur noch zu sagen, dass ich Ihnen von Herzen viel Freude mit diesem Buch und der Arbeit mit SwiftUI wünsche. Ergänzende Artikel und Videos rund um die Entwicklung für Apple-Plattformen finden Sie auf meinem Blog unter [letscode.thomassillmann.de](http://letscode.thomassillmann.de).

*Ihr Thomas Sillmann*

Aschaffenburg, August 2020