Vorwort

Objective-C ist weitschweifig. Objective-C ist plump. Objective-C ist hässlich. All diese Meinungen über Objective-C habe ich schon gehört. Ich dagegen empfinde die Sprache als elegant, flexibel und schön. Damit sie so sein kann, müssen Sie allerdings nicht nur die Grundlagen, sondern auch die Eigenheiten, Fallstricke und Feinheiten kennen, und darum geht es in diesem Buch.

Über dieses Buch

Die Syntax von Objective-C können Sie mit diesem Buch nicht lernen. Es wird vorausgesetzt, dass Sie sie bereits kennen. Stattdessen soll Ihnen dieses Buch beibringen, wie Sie das ganze Potenzial der Sprache ausschöpfen, um guten Code zu schreiben. Objective-C ist dank seiner Wurzeln in Smalltalk außerordentlich dynamisch. Ein Großteil der Arbeit, die in anderen Sprachen gewöhnlich der Compiler ausführt, wird in Objective-C von der Laufzeitkomponente erledigt. Das kann dazu führen, dass Code beim Testen hervorragend funktioniert, in der Produktion aber auf seltsame Weise fehlschlägt, etwa bei der Verarbeitung ungültiger Daten. Die beste Lösung zur Vermeidung solcher Probleme besteht natürlich darin, guten Code zu schreiben.

In vielen der Themen dieses Buches geht es streng genommen nicht um Objective-C im engeren Sinne, sondern um Einrichtungen aus den Systembibliotheken, z. B. um Grand Central Dispatch aus `libdispatch`. Es werden auch viele Klassen aus dem Foundation-Framework behandelt, nicht zuletzt auch die Wurzelklasse `NSObject`, da moderne Objective-C-Programmierung gewöhnlich für die Plattformen OS X und iOS eingesetzt wird. Bei der Entwicklung für diese Betriebssysteme werden Sie zweifellos die System-Frameworks Cocoa bzw. Cocoa Touch verwenden.

Seit dem Aufkommen von iOS haben sich Scharen von Entwicklern auf die Programmierung in Objective-C gestürzt. Manche dieser Entwickler sind sind Neulinge, andere kommen von Java oder C++, wieder andere aus der Webentwicklung. Zu welcher Gruppe Sie auch immer gehören mögen, nehmen Sie sich die Zeit, die Sprache effektiv zu lernen. Dadurch können Sie Code schreiben, der effizienter ist, sich leichter warten lässt und weniger Fehler enthält.

Ich habe nur etwa sechs Monate gebraucht, um dieses Buch zu schreiben, aber Jahre der Erfahrung sind darin eingeflossen. Aus einer Laune heraus hatte ich einen iPod touch gekauft, und als das erste SDK veröffentlicht wurde, entschloss ich mich, ein wenig mit der Entwicklung herumzuspielen. Das führte dazu, dass ich meine erste »App« schrieb, die ich unter dem Namen Subnet Calc veröffentlichte und die weit häufiger heruntergeladen wurde, als ich mir ausgemalt hatte. Damit stand für mich fest, dass meine Zukunft in dieser wunderschönen Sprache lag, die ich neu kennengelernt hatte. Seit diesem Zeitpunkt erforsche ich Objective-C und führe auf meiner Website *www.galloway.me.uk/* ein Blog darüber. Am meisten interessieren mich die internen Mechanismen, z. B. die Funktionsweise von Blöcken und ARC. Als ich die Gelegenheit erhielt, ein Buch über diese Sprache zu schreiben, ergriff ich meine Chance.

Um die Möglichkeiten dieses Buches ganz auszuschöpfen, sollten Sie darin herumstöbern und sich die Themen herauspicken, die für Sie besonders interessant sind oder die Aufgaben betreffen, an denen Sie gerade arbeiten. Alle Themen können unabhängig voneinander gelesen werden, wobei Querverweise zu verwandten Themen führen. In den einzelnen Kapiteln sind jeweils ähnliche Themen zusammengefasst, weshalb Sie sich an den Kapitelüberschriften orientieren können, um die Themen zu bestimmten Sprachmerkmalen zu finden.

Zielgruppe

Dieses Buch richtet sich an Entwickler, die ihre Kenntnisse in Objective-C vertiefen und Code schreiben möchten, der wartungsfreundlich und effizient ist und weniger Fehler enthält. Auch wenn Sie noch kein Objective-C-Entwickler sind, aber einen Hintergrund in anderen objektorientierten Sprachen wie Java oder C++ haben, können Sie hier etwas lernen. In diesem Fall ist es jedoch klug, sich zunächst eingehender mit der Syntax von Objective-C zu beschäftigen.

Der Inhalt dieses Buches

Dieses Buch dient nicht dazu, Ihnen die Grundlagen von Objective-C beizubringen, die Sie aus vielen anderen Büchern und Quellen lernen können. Stattdessen geht es darum, wie Sie diese Sprache effektiv einsetzen. Das Buch ist in Themen mit einer gut verdaulichen Menge an Informationen gegliedert, und diese Themen wiederum sind logisch zu Kapiteln geordnet:

- **Kapitel 1: Machen Sie sich mit Objective-C vertraut**
  Hier lernen Sie die wichtigsten allgemeinen Prinzipien der Sprache kennen.

- **Kapitel 2: Objekte, Nachrichten und die Laufzeit**
  Das Thema dieses Kapitels sind die Beziehungen und Wechselwirkungen zwischen den Objekten, die ein wichtiges Merkmal jeder objektorientierten Sprache bilden. Außerdem wird die Laufzeitkomponente genauer untersucht.

- **Kapitel 3: Interface- und API-Design**
  Code wird selten nur für den einmaligen Gebrauch geschrieben und dann nie wieder verwendet. Selbst wenn Sie ihn nicht veröffentlichen, werden Sie Ihren Code wahrscheinlich in mehr als einem Projekt einsetzen. In diesem Kapitel erfahren Sie, wie Sie Klassen schreiben, die in Objective-C nicht fremdkörperhaft wirken.

- **Kapitel 4: Protokolle und Kategorien**
  Protokolle und Kategorien sind wichtige Sprachmerkmale, die Sie beherrschen müssen. Ihre effektive Nutzung macht Ihren Code leichter lesbar, wartungsfreundlicher und weniger fehleranfällig. Dieses Kapitel hilft Ihnen dabei, die Dinge zu meistern.

- **Kapitel 5: Speicherverwaltung**
  Das Speicherverwaltungsmodell von Objective-C nutzt Reference Counting, was für Neulinge lange Zeit eine heikle Angelegenheit war, vor allem für diejenigen, die bereits in einer Sprache mit Garbage Collector programmiert haben. Die Einführung von Automatic Reference Counting (ARC) macht die Sache einfacher, aber Sie müssen trotzdem noch auf eine Menge wichtiger Dinge achten, damit Ihr Objektmodell korrekt ist und keine Speicherlecks aufweist. Dieses Kapitel fördert Ihre Wahrnehmung für häufig vorkommende Fallstricke der Speicherverwaltung.

- **Kapitel 6: Blöcke und Grand Central Dispatch**
  Blöcke sind lexikalische Closures für C, die von Apple eingeführt wurden. Sie werden in Objective-C häufig verwendet, um Dinge zu erreichen, für die sonst viel repetitiver Standardcode geschrieben werden müsste, und um eine Codetrennung herbeizuführen. Grand Central Dispatch (GCD) bietet eine einfache Schnittstelle für Multithreading. Blöcke werden dabei als GCD-Aufgaben betrachtet, die je nach verfügbaren Systemressourcen auch parallel ausgeführt werden können. Die Lektüre dieses Kapitels versetzt Sie in die Lage, diese beiden wichtigen Technologien anzuwenden.

- **Kapitel 7: Die System-Frameworks**
  Objective-C-Code schreiben Sie gewöhnlich für OS X oder iOS. In diesen Fällen steht Ihnen mit `Cocoa` bzw. `Cocoa Touch` der komplette Vorrat an System-Frameworks zur Verfügung. Dieses Kapitel gibt einen kurzen Überblick über die Frameworks und stellt einige ihrer Klassen genauer vor.

Wenn Sie irgendwelche Fragen, Kommentare oder Bemerkungen zu diesem Buch haben, können Sie sich gern (in englischer Sprache) an mich wenden.

Kontaktinformationen finden Sie auf der Website zu diesem Buch auf *www.effectiveobjectivec.com*.