

1 Einführung

Objective-C ist eine objektorientierte Programmiersprache. Sie wurde Anfang der 80er-Jahre von Brad Cox und Tom Love von der Firma *Stepstone* entwickelt und ist stark an die heute fast vergessene Sprache *Smalltalk* angelehnt. Später gelangte sie durch die Akquise Steve Jobs' ehemaliger Firma *NEXTStep* zu Apple. Bei Apple wird Objective-C in allen höheren Systemschichten eingesetzt und ist insbesondere in der Programmierung von Endnutzer-Anwendungen nicht zu umgehen. Zumindest laut dem TIOBE-Index für die populärsten Programmiersprachen [20] hat Objective-C mittlerweile Sprachen wie *Ruby*, *JavaScript*, *Perl* und *Python* abgehängt und rangiert jetzt ungefähr auf derselben Popularitätsebene wie *C#* und *PHP* – ein Big Player also.

In diesem Buch wird der Versuch unternommen, die Sprache mehr oder weniger aus dem Nichts zu konstruieren und sie bis ins kleinste Detail auseinanderzunehmen. Es ist ein Buch mit nützlichem Detailwissen für den täglichen Gebrauch, das auch als Referenz und Nachschlagewerk dienen kann. Jedes Feature und jedes Konzept wird ausführlich besprochen und mit Beispielen illustriert.

Man wird hier allerdings nicht lernen, was objektorientierte Programmierung ist oder wie man Programme für das iPhone entwickelt. Dieses Buch richtet sich ganz klar an Umsteiger anderer objektorientierter Sprachen und an Entwickler, die bereits Erfahrungen mit Objective-C gemacht haben. So werden zum besseren Verständnis an vielen Stellen Vergleiche zu den weitverbreiteten Sprachen C++ und Java gezogen. Eine gewisse Tendenz, Objective-C besser als alle anderen Sprachen zu finden, sei dem Autor dabei bitte verziehen.

Wie schon erwähnt sollte ein Leser bereits Erfahrungen mit der objektorientierten Programmierung haben. Konzepte wie Klassen und Objekte sollten ebenso fest verankert sein wie das Wissen, was ein Compiler ist und wofür er benutzt wird. Es sollte klar sein, dass es einen Unterschied zwischen der Laufzeit eines Programms und der Compilezeit gibt.

Weil Objective-C »nur« eine Erweiterung der prozeduralen Sprache C darstellt, ist eine grundlegende Kenntnis dieser Sprache ebenfalls klar von Vorteil. Insbesondere in den Kapiteln 5 und 10 wird allgemeines

Fokus

Voraussetzungen

Grundwissen über Zeiger, Funktionszeiger, Referenzierung/Dereferenzierung und den Unterschied zwischen Stack und Heap vorausgesetzt. An den betreffenden Stellen wird es aber auch fast immer eine Referenz zur Auffrischung der erforderlichen Kenntnisse geben.

Syntax

Zuletzt soll gleich im Voraus auf die etwas merkwürdige Syntax hingewiesen werden. Objective-C existiert friedlich neben den Sprachen C und C++ und lässt sich problemlos mit ihnen mischen. Das bedeutet aber auch, dass die Sprachkonstrukte jeweils andere sein müssen. Die meisten Schlüsselwörter beginnen daher mit dem sonst ungenutzten -Zeichen und so manchem Attribut ist der C-übliche doppelte Unterstrich `--` vorangestellt. Am auffälligsten sind aber sicher die Aufrufe von Methoden (in Objective-C eigentlich »Nachrichten« genannt): Statt der in fast allen objektorientierten Sprachen üblichen runden Klammern () verwendet Objective-C eckige []. Auch steht der Name einer aufgerufenen Methode in Objective-C zwischen den einzelnen Parametern. Bei einem Neuling kann das sicher erst einmal zu etwas Verwirrung führen – mit der Zeit gewöhnt man sich aber daran und erkennt die Vorteile. Dazu aber mehr in den nächsten Kapiteln.

Objective-C ist sowieso etwas anders als die meisten anderen Sprachen. Nicht nur die Syntax, auch die Konzepte weichen zum Teil erheblich von der Norm ab. Alles ist dynamisch und noch zur Laufzeit veränderbar. Das schafft große Möglichkeiten, von denen ein paar auch vorgestellt werden. Ich wünsche viel Spaß beim Lesen!