



Sebastian
Meyer

Torben
Wichers

2. Auflage

Objective-C 2.0

Programmierung für Mac OS X und iPhone

Inhaltsverzeichnis

Danksagung	13
Über die Autoren	13
Einleitung	15
An wen richtet sich das Buch?	16
Aufbau des Buches	17
Beispiele und verwendete Werkzeuge	21
Typografische Konventionen	22
Historische Einordnung	23
Objective-C als Erweiterung zu ANSI-C	27
Teil I Die Programmiersprache Objective-C 2.0	33
I Objekte und Klassen	35
I.1 Was ist objektorientierte Programmierung?	35
I.2 Objekte und Dynamic Typing	37
I.3 Nachrichten und Dynamic Binding	39
I.4 Klassen	44
I.4.1 Static Typing als Alternative zum Dynamic Typing	46
I.4.2 Aufbau einer Klasse	47
I.4.3 Vererbung	51
I.4.4 Instanzvariablen	54
I.4.5 Methoden und Klassenmethoden	60
I.5 Erzeugung von Objekten	66
I.5.1 Initialisierungsmethoden	68
I.5.2 Convenience-Konstruktoren	71
I.6 Übungsaufgaben	74
2 Wie werden Nachrichten verarbeitet?	77
2.1 Übersetzung von Methodenaufrufen	77
2.2 Struktur einer Methodenimplementierung	80
2.3 Auflösung der dynamischen Bindung	82

2.4	Forward Invocation	85
2.5	Übungsaufgaben	89
3	Kategorien	91
3.1	Erweiterung durch Kategorien	91
3.2	Verstecken von Methoden.	94
3.3	Kategorien und die Klasse »NSObject«	96
3.4	Extensions	98
3.5	Übungsaufgaben	100
4	Properties	103
4.1	Zugriffsmethoden	104
4.2	Deklarative Zugriffsmethoden	105
4.2.1	Synthetische Properties	106
4.2.2	Attribute für Properties	108
4.2.3	Eigene Implementierungen von Properties	112
4.2.4	Neudeklaration von Properties	114
4.3	Der Punkt-Operator.	115
4.4	Übungsaufgaben	118
5	Protokolle	121
5.1	Protokolle und Hierarchien	122
5.2	Verwendung von Protokollen	124
5.3	Grenzen von Protokollen	131
5.4	Optionale Protokollmethoden.	133
5.5	Informelle Protokolle	140
5.6	Übungsaufgaben	141
6	Ausnahmebehandlung	143
6.1	Auslösen von Ausnahmen	145
6.2	Fangen von Ausnahmen	150
6.3	Übungsaufgaben	156
7	Blöcke	159
7.1	Zugriff auf Variablen.	163
7.2	Blöcke in den Heap kopieren	166
7.3	Blöcke und Objective-C.	169
7.4	Wie werden Blöcke realisiert?.	172
7.5	Übungsaufgaben	185

Teil II	Objective-C-Programmierung für Mac OS X und iPhone OS	187
8	Die Cocoa-Umgebung	189
8.1	Das Cocoa-Framework	191
8.1.1	Das Foundation Framework	192
8.1.2	Application Kit	195
8.2	Ein erstes Beispielprogramm mit dem Cocoa-Framework	197
8.2.1	Ein neues Projekt in Xcode	198
8.2.2	Eine Anwendung mit grafischer Oberfläche	201
8.3	Cocoa Touch unter iOS	208
8.3.1	Die erste iPhone-Anwendung	210
8.3.2	Ausführung auf dem iPhone	215
8.4	Übungsaufgaben	219
9	Memory Management	223
9.1	Erzeugen von neuen Objekten	229
9.2	Kopieren von Objekten	231
9.3	Reference Counting	234
9.3.1	Besitz und Freigabe von Objekten	236
9.3.2	Autorelease Pools	245
9.3.3	Die dealloc-Methode	251
9.3.4	Properties, Getter und Setter	253
9.3.5	Zyklische Abhängigkeiten	255
9.3.6	Implementierung von Reference Counting	262
9.4	Garbage Collection	268
9.4.1	Barrieren und Generationen	274
9.4.2	Die finalize-Methode	275
9.4.3	Schwache Referenzen	280
9.5	Blöcke und Memory Management	282
9.6	Übungsaufgaben	285
10	Grundlegende Klassen	287
10.1	Byteorientierte Daten	288
10.2	Die Klasse NSCoder	293
10.3	Zeichenketten	297
10.4	Zeit- und Datumswerte	309
10.4.1	Zeitpunkte und -intervalle	309
10.4.2	Kalender	311

10.4.3	Zeitzone	315
10.4.4	Ausgabe und Erkennung von Zeitwerten	317
10.5	Kapselung primitiver Datentypen	321
10.6	Ausnahmen mit NSError	326
10.7	Fehler mit NSError	327
10.8	Übungsaufgaben	332
II	Collections	333
II.1	Arrays	334
II.1.1	Unveränderliche Arrays	334
II.1.2	Veränderliche Arrays	337
II.1.3	Sortieren von Arrays	343
II.1.4	Filtern von Arrays	350
II.2	Sets	355
II.2.1	Unveränderliche Sets	355
II.2.2	Veränderliche Sets	357
II.2.3	Beziehungen zwischen Sets	358
II.2.4	Filtern von Sets mittels Blöcken	361
II.2.5	Counted Sets	363
II.3	Dictionaries	364
II.3.1	Unveränderliche Dictionaries	365
II.3.2	Veränderliche Dictionaries	367
II.3.3	Sortieren von Dictionaries	370
II.3.4	Filtern von Dictionaries mittels Blöcken	371
II.4	Durchlaufen von Collections	373
II.4.1	Durchlaufen mittels NSEnumerator	373
II.4.2	Durchlaufen mittels Blöcken	375
II.4.3	Durchlaufen veränderlicher Collections	379
II.5	Übungsaufgaben	380
12	Eingabe und Ausgabe	383
12.1	Verzeichnisse und Dateien	383
12.2	Operationen im Dateisystem	390
12.3	Streams	397
12.3.1	InputStreams	399
12.3.2	OutputStreams	401
12.3.3	Nicht-blockierender Zugriff auf Streams	404
12.4	Höhere Ebenen der Ein- und Ausgabe	412
12.5	Übungsaufgaben	422

13	Introspektion	425
13.1	Introspektion über NSObject	426
13.2	Introspektion mit der Runtime Library	430
13.3	Metaprogrammierung	443
13.4	Übungsaufgaben	453
14	Design Patterns für Objective-C	457
14.1	Schnelle Iteration	458
14.1.1	Vorteile der schnellen Iteration	461
14.1.2	Das Protokoll NSFastEnumeration	461
14.2	Key-Value Coding	470
14.2.1	Terminologie des Key-Value Codings	472
14.2.2	Zuweisen und Auslesen von Werten	473
14.2.3	Implementierung von 1:m-Beziehungen	480
14.2.4	Skalare Typen und Strukturen	483
14.2.5	Key-Value-Validierung	486
14.2.6	Performance	490
14.3	Observer-Pattern	491
14.3.1	Key-Value Observing	492
14.3.2	Benutzung eines NotificationCenter	503
14.4	Delegation	516
14.4.1	Delegates zur Auslagerung von Funktionalität	516
14.4.2	Delegates als Datenquelle	519
14.5	Singletons	521
14.5.1	Ein einfaches Singleton	523
14.5.2	Singletons und Threading	527
14.6	Übungsaufgaben	530
15	Threading	533
15.1	Operationen	535
15.1.1	Parallelisierung von Operationen	538
15.1.2	Priorisierung von Operationen	542
15.1.3	Abhängigkeiten zwischen Operationen	544
15.1.4	Die Klasse NSInvocationOperation	545
15.1.5	Die Klasse NSBlockOperation	546
15.2	Threads	547
15.3	Synchronisation von Threads	556
15.3.1	Nicht-blockierende Synchronisation	558
15.3.2	Locks	562
15.3.3	Conditions	568

15.4	Run Loops	569
15.5	Run-Loop-Quellen	579
15.5.1	Timer-Quellen	580
15.5.2	Perform-Selektor-Quellen	585
15.5.3	Port-basierte Quellen	589
15.5.4	Benutzerdefinierte Quellen	602
15.6	Übungsaufgaben	605
16	Bundles	609
16.1	Anatomie eines Bundles	610
16.2	Lokalisierung	610
16.3	Verwendung von Bundles	617
16.4	Übungsaufgaben	624
Teil III Anhänge		627
A	Lösungsvorschläge	629
A.1	Objekte und Klassen	629
A.2	Wie werden Nachrichten verarbeitet?	632
A.3	Kategorien	636
A.4	Properties	637
A.5	Protokolle	638
A.6	Ausnahmebehandlung	640
A.7	Blöcke	641
A.8	Die Cocoa-Umgebung	644
A.9	Memory Management	648
A.10	Grundlegende Klassen	651
A.11	Collections	653
A.12	Eingabe und Ausgabe	657
A.13	Introspektion	661
A.14	Design Patterns für Objective-C	662
A.15	Threading	667
A.16	Bundles	674

B	Eine kurze C-Einführung	677
B.1	Aufbau eines C-Programms	677
	B.1.1 Kommentare	680
	B.1.2 Basisdatentypen und Konstanten	681
	B.1.3 Funktionen	685
	B.1.4 Variablen	691
	B.1.5 Kompilieren mehrerer Dateien	696
B.2	Der C-Präprozessor	698
	B.2.1 Definition von Makros	698
	B.2.2 Einfügen von Dateien	702
	B.2.3 Bedingte Übersetzung	706
	B.2.4 Vordefinierte Namen	708
B.3	Anweisungen und Kontrollstrukturen	709
	B.3.1 Bedingte Anweisungen	710
	B.3.2 Schleifen	713
	B.3.3 Sprunganweisungen	715
B.4	Ausdrücke und Operatoren	716
	B.4.1 Arithmetische Operatoren	718
	B.4.2 Vergleichsoperatoren	718
	B.4.3 Bit-Manipulationen	719
	B.4.4 Wertzuweisungen	719
	B.4.5 Logische Operatoren	721
	B.4.6 Typ-Operatoren	722
	B.4.7 Operatorprioritäten	723
B.5	Weitere Datentypen	724
	B.5.1 Pointer	725
	B.5.2 Arrays	728
	B.5.3 Strukturen	731
	Stichwortverzeichnis	735