

Fritz Anderson

Xcode 3

Übersetzung aus dem Amerikanischen
von Reinhard Engel



mitp

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.

ISBN 978-3-8266-5518-0

1. Auflage 2009

Alle Rechte, auch die der Übersetzung, vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder einem anderen Verfahren) ohne schriftliche Genehmigung des Verlages reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden. Der Verlag übernimmt keine Gewähr für die Funktion einzelner Programme oder von Teilen derselben. Insbesondere übernimmt er keinerlei Haftung für eventuelle aus dem Gebrauch resultierende Folgeschäden.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Authorized translation from the English language edition, entitled XCODE 3 UNLEASHED, 1st Edition, 0321552636 by ANDERSON, FRITZ, published by Pearson Education, Inc, publishing as Sams Publishing, Copyright © 2009 Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc. GERMAN language edition published by mitp-Verlag, Verlagsgruppe Hüthig Jehle Rehm GmbH, Copyright © 2009.

Printed in Austria

© Copyright 2009 by mitp-Verlag

Verlagsgruppe Hüthig Jehle Rehm GmbH

Heidelberg, München, Landsberg, Frechen, Hamburg

www.it-fachportal.de

Lektorat: Sabine Schulz

Korrektorat: Petra Heubach-Erdmann

Satz: III-satz, Husby, www.drei-satz.de

Inhaltsverzeichnis

	Einführung	17
Teil I	Der Lebenszyklus einer Mac-OS-X-Anwendung	29
I	Der Einstieg	31
1.1	Xcode starten	31
1.2	Hello, World.	33
1.3	Was wurde wo gespeichert?	38
1.4	Zusammenfassung	39
2	Einfacher Workflow und passives Debugging	41
2.1	Eine lineare Regression berechnen	42
2.2	Aktionsplan	43
2.3	Ein Command-Line-Tool	44
2.4	Build-Fehler	45
2.5	Einfaches Debugging	50
2.6	Zusammenfassung	51
3	Einfaches aktives Debuggen	53
3.1	Der nächste Schritt	53
3.2	Aktives Debugging	54
3.3	Zusammenfassung	62
4	Grundlagen der Kompilierung	63
4.1	Kompilierung	63
4.2	Linking	66
4.3	Dynamic Loading	68
4.4	Legacy-Technologien	69
4.4.1	Prebinding	69
4.4.2	ZeroLink	69
4.5	Zusammenfassung	70

5	Einstieg in eine Cocoa-Anwendung	71
5.1	Vorgehensplan	71
5.1.1	Programmaufgaben	71
5.1.2	Model-View-Controller	71
5.1.3	Das Modell	72
5.1.4	Der Controller	74
5.1.5	Die Views	75
5.2	Ein neues Projekt starten	76
5.2.1	Linrg hinzufügen	76
5.3	Implementation: Model	80
5.3.1	Die Model-Klasse DataPoint	80
5.3.2	Die Model-Klasse Regression	84
5.3.3	Fertig	88
5.4	Zusammenfassung	88
6	Eine Cocoa-Anwendung: Views	89
6.1	Interface Builder	89
6.1.1	Layout	92
6.1.2	Größenänderung	95
6.1.3	Eine Split-View	99
6.2	Zusammenfassung	101
7	Eine Cocoa-Anwendung: Controller	103
7.1	Der nächste Schritt	103
7.2	Einen Controller hinzufügen	106
7.2.1	Der Objekt-Controller Document	108
7.2.2	Der Objekt-Controller Model	109
7.2.3	Der Array-Controller DataPoints	111
7.3	Werte-Bindung	111
7.4	Actions und Outlets	114
7.5	MyDocument	116
7.6	Anwendungs-Properties	119
7.7	Building	121
7.8	Zusammenfassung	122
8	Versionskontrolle	123
8.1	Ein Repository einrichten	124
8.2	Das Home-Verzeichnis vorbereiten	125
8.3	Das Repository mit Xcode verknüpfen	128
8.4	Linear der Versionskontrolle unterstellen	130
8.5	Linear mit Subversion kontrollieren	134
8.6	Mit Revisionen arbeiten	136
8.7	Rollbacks	146
8.8	Tagging	147
8.9	Zusammenfassung	149

9	Property Lists	151
9.1	Datentypen	151
9.2	Property-List-Dateien	152
9.2.1	Eine Property List schreiben	152
9.3	Property Lists untersuchen	162
9.3.1	Als Text	162
9.3.2	Der Property List Editor	164
9.4	Andere Formate	166
9.4.1	ASCII Property Lists	166
9.4.2	Binary Property Lists	167
9.5	Text-Makros	167
9.6	Zusammenfassung	174
10	Libraries und Dependent Targets	175
10.1	Ein Target hinzufügen	175
10.2	Library-Design	177
10.3	Linear ändern	180
10.4	Ein abhängiges Target	182
10.5	Die Library untersuchen	183
10.6	Die Library ausführen	186
10.7	Zusammenfassung	186
11	File Packages und Bundles	187
11.1	Ein einfaches Package: RTFD	188
11.2	Bundles	189
11.3	Anwendungs-Bundles	190
11.4	Die Datei Info.plist	191
11.4.1	Keys für alle Bundles	192
11.4.2	Keys für Anwendungen	193
11.4.3	Keys für Plugins	196
11.4.4	Keys für Java	196
11.4.5	Keys für Preference Panes	197
11.4.6	Keys für Dashboard Widgets	197
11.5	Zusammenfassung	198
12	Unit-Testing	199
12.1	Ein Unit-Test-Target hinzufügen	199
12.2	Refactoring	203
12.3	Die Tests ausführen	207
12.4	Ein Dependent Test	209
12.5	Architekturübergreifend arbeiten	212
12.6	Zusammenfassung	212
13	Eine anwendungsspezifische View erstellen	213
13.1	Controller	213
13.2	View	215
13.3	Das Delegate-Design-Pattern	220

13.4	Die anwendungsspezifische View	222
13.5	Das Fenster anzeigen	228
13.6	Testen	229
13.7	Eine View debuggen	231
13.8	Zusammenfassung	234
14	Dynamic Libraries und Frameworks	235
14.1	Ein Framework-Target hinzufügen	235
14.1.1	Info.plist	236
14.1.2	Dateien zuweisen	236
14.2	Framework-Struktur	242
14.3	Das Framework benutzen	243
14.4	Wo Frameworks gespeichert werden	246
14.4.1	Ein Public Framework	248
14.4.2	Ein Private Framework	249
14.5	Zusammenfassung	252
15	Dokumentation in Xcode	253
15.1	HeaderDoc	253
15.2	User Scripts in Xcode	259
15.3	DocSets	263
15.3.1	Den Workspace vorbereiten	264
15.3.2	Das Mindeste, was wir tun können	265
15.4	Shell Script Targets	268
15.5	Browsing	270
15.6	API-Dokumentation	272
15.7	Zusammenfassung	276
16	Die Datenmodellierungswerkzeuge verwenden	277
16.1	Datenmodellierung	279
16.2	Das Modell überarbeiten	282
16.2.1	DataPoint.h	282
16.2.2	DataPoint.m	283
16.2.3	Initialisierer	284
16.2.4	Accessors	284
16.2.5	Regression	285
16.2.6	MyDocument	289
16.3	Interface Builder	291
16.4	Erstellen und ausführen	293
16.5	Eine Entity hinzufügen	294
16.5.1	Das Datenmodell erweitern	294
16.5.2	Benutzerschnittstelle	295
16.5.3	Erste Ausführung	299
16.5.4	Eine weitere Tabelle	300
16.6	Zusammenfassung	302

17	Cross-Development	303
17.1	Cross-Development-SDKs	303
17.2	Weak Linking	306
17.3	NIB-Kompatibilität	307
17.4	Universal Binaries	310
	17.4.1 Linear prüfen	312
	17.4.2 Testen	313
17.5	Erstellung für verschiedene Betriebssysteme und Prozessoren	313
17.6	Projekt-Kompatibilität	315
17.7	Zusammenfassung	316
18	Spotlight (oder wie man ein Plugin erstellt)	317
18.1	Wie Spotlight seine Daten bekommt	317
18.2	Uniform Type Identifiers	320
18.3	Das Spotlight-Plugin	322
	18.3.1 Das MetaLinear-Projekt	323
	18.3.2 MetaLinear-Projektdateien	324
	18.3.3 Packaging des Plugins	330
	18.3.4 Unsere Arbeit prüfen	330
18.4	Core Data und Metadaten	331
18.5	Der Praxisbeweis	334
18.6	Zusammenfassung	336
19	Letzte Hand anlegen	337
19.1	Die Menüs anpassen	337
	19.1.1 Singularität vermeiden	338
19.2	Lokalisierung	340
	19.2.1 Credits.rtf	340
	19.2.2 MainMenu.xib	341
	19.2.3 MyDocument.xib	342
	19.2.4 GraphWindow.xib	344
	19.2.5 InfoPlist.strings	344
	19.2.6 Testlauf	346
	19.2.7 Localizable.strings	348
19.3	Den Speicherverbrauch prüfen	350
19.4	Instruments	352
	19.4.1 Verwaiste Blöcke	354
	19.4.2 Call Trees mit Speicherlecks	355
	19.4.3 Das ObjectAlloc-Instrument	356
	19.4.4 Das Instruments-Dokument	358
	19.4.5 Benutzeraktionen aufzeichnen	358
19.5	Die Release-Build-Konfiguration	361
19.6	Das Produkt verschlanken	363
	19.6.1 Die Symboltabelle strippen	364
	19.6.2 Dead Code Stripping	365
19.7	Zusammenfassung	366

Teil II Xcode Tasks

367

20	In einem Xcode-Projekt navigieren	369
20.1	Editor-Panes	369
20.1.1	Code Sense	370
20.1.2	Sprünge	371
20.1.3	Die Navigationsleiste	372
20.1.4	Code Focus	374
20.1.5	Die Debugger-Leiste	375
20.1.6	Editor-Modi	375
20.2	Das Fenster Project Find	376
20.3	Die Favoritenleiste	378
20.4	Die Liste Groups & Files	379
20.4.1	Die Projektgruppe	379
20.5	Die Targets-Gruppe	381
20.5.1	Die Executables-Gruppe	382
20.5.2	Smart Groups	383
20.5.3	Symbol Smart Group	385
20.6	Das Fenster Class Browser	385
20.7	Class Modeler	386
20.8	Projekt-Layout	388
20.8.1	Default-Layout	388
20.8.2	All-in-One-Layout	390
20.8.3	Condensed-Layout	391
20.9	Zusammenfassung	392
21	Xcode für make-Veteranen	393
21.1	Build-Variablen von Xcode	395
21.2	Anwendungsspezifische Build-Regeln	396
21.3	Build-Phase Run Script	397
21.4	Unter der Haube: ein einfacher Build	399
21.4.1	Copy Structural Files	400
21.4.2	XIB kompilieren	400
21.4.3	Skript ausführen	401
21.4.4	Bundle-Ressourcen kopieren	401
21.4.5	Sourcen kompilieren	402
21.4.6	Linkage (erste Architektur)	404
21.4.7	Sourcen kompilieren (zweite Architektur)	404
21.4.8	Linkage (zweite Architektur)	405
21.4.9	Datenmodelle kompilieren	405
21.4.10	Universal Binary erstellen	405
21.4.11	Abschließendes Touch	406
21.5	Das Tool xcodebuild	406
21.6	Hierarchie der Einstellungen	407
21.7	Build-Konfigurationen	408
21.8	Zusammenfassung	412

22	Mehr über das Debuggen	413
22.1	Debugging-Techniken	413
22.2	Werte drucken	417
22.3	Anwendungsspezifische Formatierer	419
22.4	Breakpoint-Befehle	421
22.5	Breakpoint Conditions	423
22.6	Lazy Symbol Loading	424
22.7	Zombies	425
	22.7.1 Released-Pointer Aliasing	426
	22.7.2 NSZombieEnabled	428
22.8	Der Mini-Debugger und der In-Editor-Debugger	429
22.9	Datatypes	432
22.10	Zusammenfassung	433
	22.10.1 Weitere Literatur	433
23	Xcode und Geschwindigkeit	435
23.1	Vorkompilierte Header	435
23.2	Predictive Compilation	436
23.3	Distributed Builds	436
	23.3.1 Alles über Distributed Builds	437
	23.3.2 Shared Workgroup Builds	438
	23.3.3 Dedicated Network Builds	439
23.4	Projektindizierung	439
23.5	Zusammenfassung	441
24	Ein Legacy-Projekt	443
24.1	Das Projekt vorbereiten	444
24.2	Der Organizer	445
	24.2.1 Die Files-Liste	445
	24.2.2 Die Organizer-Toolbar	447
	24.2.3 Konfigurieren und erstellen	450
	24.2.4 Installation	451
	24.2.5 Ausführen	453
24.3	Ein External-Build-System-Projekt	454
	24.3.1 Code Sense ist da	457
	24.3.2 Ausführung	458
	24.3.3 Debugging	459
	24.3.4 Die Grenzen des External Build Systems	459
24.4	Zusammenfassung	460
25	Shark und die CHUD-Tools	461
25.1	Shark	461
	25.1.1 Das Problem	461
	25.1.2 Shark starten	462
	25.1.3 Analyse	463
	25.1.4 Die View Top-Down	465

25.1.5	Data-mining im Call Stack	465
25.1.6	Messen, dann optimieren	468
25.1.7	Die Wirkung	470
25.2	Die anderen Performance-Tools	472
25.2.1	BigTop	472
25.2.2	Reggie SE	473
25.2.3	SpindownHD	473
25.2.4	Saturn	473
25.2.5	MallocDebug	473
25.2.6	ObjectAlloc und Sampler	473
25.2.7	Quartz Debug	474
25.2.8	Spin Control	476
25.2.9	Thread Viewer	476
25.2.10	CHUD Remover	477
25.3	Zusammenfassung	477
26	Instruments	479
26.1	Was ist Instruments?	479
26.2	Instruments ausführen	480
26.3	Das Fenster Trace Document	480
26.3.1	Die Toolbar	482
26.3.2	Die Track-Pane	483
26.3.3	Die Detail-Pane	484
26.3.4	Die Pane Extended Detail	486
26.3.5	Controls	487
26.4	Die Library	487
26.5	Ein Instrument ausführen	488
26.5.1	Ein Instrument konfigurieren	488
26.5.2	Aufzeichnen	490
26.5.3	Speichern und erneut öffnen	491
26.6	Die Instrumente	492
26.6.1	Core Data	492
26.6.2	File System	493
26.6.3	Garbage Collection	494
26.6.4	Graphics	494
26.6.5	Input/Output	494
26.6.6	Master Tracks	494
26.6.7	Memory	495
26.6.8	System	496
26.6.9	Threads/Locks	498
26.6.10	User Interface	498
26.7	Custom Instruments	498
26.8	Die Templates	500
26.9	Zusammenfassung	502

27	Ergänzende Themen.	503
27.1	Verschiedene Stolpersteine.	503
27.2	Verschiedene Tipps	506
27.3	Mehr Dokumentation	512
27.3.1	Updates der Dokumentation	512
27.3.2	Boolesche Textsuchen.	513

Teil III Anhänge		515
-------------------------	--	------------

A	Einige Build-Variablen	517
A.1	Nützliche Build-Variablen.	518
A.1.1	Umgebung (Environment)	519
A.1.2	Build-Targets	520
A.1.3	Source-Verzeichnisse (Quell-Verzeichnisse)	521
A.1.4	Destination-Verzeichnisse (Ziel-Verzeichnisse)	521
A.1.5	Bundle-Verzeichnisse	522
A.1.6	Compiler-Einstellungen	522
A.1.7	Suchpfade	523
A.1.8	Deployment	524
A.2	Source Trees.	524
B	Projekt- und Target-Templates	527
B.1	Projekt-Templates	527
B.1.1	Application.	527
B.1.2	Audio Units	530
B.1.3	Automator Action	530
B.1.4	Bundle	531
B.1.5	Command Line Utility	531
B.1.6	Dynamic Library	531
B.1.7	Framework.	532
B.1.8	Java	532
B.1.9	Kernel Extension	533
B.1.10	Standard Apple Plug-Ins.	533
B.1.11	Static Library	535
B.1.12	Others.	535
B.2	Target-Templates	536
B.2.1	BSD	537
B.2.2	Carbon	537
B.2.3	Cocoa	538
B.2.4	Kernel Extension	539
B.2.5	Ruby	539
B.2.6	Other	539

C	Andere Ressourcen	541
C.1	Bücher	541
C.2	Aus dem Xcode-Dokumentation-Fenster	542
C.3	Im Internet	542
	C.3.1 Mailing Lists	542
	C.3.2 Usenet	542
C.4	Websites und Logs	543
C.5	Persönliche Kontakte	544
C.6	Texteditoren	544
C.7	Tools	544
	Stichwortverzeichnis	547