

Klaus M. Rodewig
Clemens Wagner

Aktuell zu iOS 5
und Xcode 4.2



```
NSFileManager *fileManager = [NSFileManager defaultManager];  
NSArray *directories = NSSearchPathForDirectoriesInDomains(NSDocumentDirectory, NSUserDomainMask, YES);  
NSString *path = [directories objectAtIndex:0];  
NSURL *url = [[results objectAtIndex:indexPath row] objectAtIndex:0];  
  
if ([[source selectedSegmentIndex] == 1]) {  
    [fileManager setUbiquitous:NO itemAtURL:url  
                destinationURL:[NSURL URLWithString:]]
```



Apps entwickeln für iPhone und iPad

Das Praxisbuch

- ▶ Grundlagen der Anwendungsentwicklung mit dem SDK 5
- ▶ Apps entwickeln, testen, absichern und veröffentlichen
- ▶ Inkl. Xcode 4, Debugging, Versionierung, zahlreicher Praxisbeispiele

LEHR-
Programm
gemäß
§14
JuSchG



Alle Beispiele, Tools sowie
Videocasts zur iPhone- und
iPad-Entwicklung

Galileo Computing



Inhalt

Geleitwort des Gutachters	13
Vorwort	15

1 Einführung 17

1.1	iOS	18
1.1.1	Core OS	21
1.1.2	Core Services	21
1.1.3	Media	23
1.1.4	Cocoa Touch	23
1.2	iOS-SDK	25
1.2.1	Xcode	27
1.2.2	Instruments	29
1.2.3	Simulator	30
1.2.4	Test auf dem iPhone	31
1.3	iOS-Hardware	32
1.4	Besonderheiten der Programmierung	36
1.4.1	Speicher	37
1.4.2	Akkulaufzeit	38
1.4.3	Benutzerschnittstelle	39
1.4.4	Sicherheit	42
1.4.5	Der App Store	43
1.5	Objektorientierte Programmierung	44
1.5.1	Objekte	46
1.5.2	Vererbung	47
1.5.3	Nachrichten	52
1.5.4	Objective-C	53

2 Einstieg in die Praxis 61

2.1	Hefte raus, Klassenarbeit! – Die Arbeit mit Klassen	67
2.1.1	Objekte erzeugen: alloc und init	70
2.1.2	Objekte löschen: release und dealloc	72
2.1.3	Initializer	75
2.1.4	Accessoren	78
2.1.5	Eigene Methoden	81
2.1.6	Vererbung	84
2.1.7	Kategorien	89
2.1.8	Protokolle	93

2.2	Das erste iOS-Projekt	94
2.2.1	Der Interface Builder	97
2.2.2	Delegation	102
2.3	Model, View und Controller	108
2.3.1	Der Aufbau des Musters	109
2.3.2	Key-Value-Observing	111
2.3.3	Target-Action-Mechanismus	113
2.3.4	Controller in Cocoa Touch	114
2.4	Speicherverwaltung, Propertyts und Key-Value-Coding	114
2.4.1	Referenzzahlen in Objective-C	116
2.4.2	Die Speicherverwaltungsregeln	119
2.4.3	Propertyts	124
2.4.4	Key-Value-Coding	129
2.4.5	Abschließende Bemerkungen	131
2.5	Automatisches Referenzzahlen	133
2.5.1	Zurück in die Zukunft	133
2.5.2	Weakie und die starken Zeiger	134
2.5.3	Autoreleasepools	137
2.5.4	Einzelgänger	138
2.5.5	Migration bestehender Projekte	140

3 Aufwachen – analoger Wecker 145

3.1	Eigene View-Klassen in Cocoa Touch	145
3.1.1	Zeichnen in Cocoa Touch	151
3.1.2	Zeitberechnung	159
3.1.3	View-Erzeugung über NIB-Dateien	161
3.1.4	Aktualisierung der Zeitanzeige	163
3.1.5	Wiederverwendbarkeit von Views	166
3.2	Views und Viewcontroller	167
3.2.1	Outlets	168
3.2.2	View-Hierarchien	175
3.2.3	Actions	176
3.2.4	Ereignisse	179
3.2.5	Controlzustände und Buttons	182
3.2.6	Direkte Gestenverarbeitung	185
3.2.7	Übergänge	193
3.2.8	Der Lebenszyklus eines Viewcontrollers	195
3.3	Lokale Benachrichtigungen	199
3.3.1	Benachrichtigungen versenden	200
3.3.2	... und verarbeiten	202

3.4	Eine App für alle	206
3.4.1	Das Retina Display	208
3.4.2	Sprachkursus für die App	210
3.4.3	Es funktioniert nicht	213
3.4.4	Universelle Apps	213
3.5	Fehlersuche	215
3.5.1	Logging	216
3.5.2	Der Debugger	217
3.5.3	Laufzeitfehler	221
3.6	Die App auf einem Gerät testen	222
3.6.1	Das iOS Developer Program	224
3.6.2	Entwicklungszertifikat und Entwicklungsprofile	224
3.6.3	Profilprobleme	229

4 Daten, Tabellen und Controller 231

4.1	Benachrichtigungen	231
4.1.1	Benachrichtigungen empfangen	231
4.1.2	Eigene Benachrichtigungen verschicken	233
4.2	Core Data	234
4.2.1	Datenmodellierung	235
4.2.2	Implementierung von Entitätstypen	239
4.2.3	Einbindung von Core Data	240
4.2.4	Der Objektkontext	242
4.2.5	Die Nachrichten des Objektkontexts	243
4.2.6	Anlegen und Ändern von Entitäten in der Praxis	244
4.3	View-Rotationen	246
4.3.1	Flexible Views dank der Autoresizingmask	247
4.3.2	Lang lebe das Rotationsprinzip	249
4.3.3	Anpassung des Layouts	251
4.4	Texte, Bilder und Töne verwalten	253
4.4.1	Die Tastatur betritt die Bühne	254
4.4.2	Fotos aufnehmen	259
4.4.3	Töne aufnehmen und abspielen	267
4.5	Tableviews und Core Data	278
4.5.1	Tableviews	279
4.5.2	Tabellenzellen gestalten	281
4.5.3	Buttons in Tabellenzellen	286
4.5.4	Zellen löschen	287
4.5.5	Core Data II: Die Rückkehr der Objekte	287
4.5.6	Prädikate	289

4.5.7	Aktualisierung des Tableviews	292
4.5.8	Das Delegate des Fetchedresultscontrollers	294
4.5.9	Tabelleneinträge suchen	298
4.6	Containerviewcontroller	301
4.6.1	Der Navigationcontroller	301
4.6.2	SplitviewController	309
4.6.3	Der PageviewController	313
4.6.4	Subview- und Containerviewcontroller	320

5 Animationen und Layer 325

5.1	Modell und Controller	326
5.1.1	iOS Next Topmodel	326
5.1.2	View an Controller	329
5.1.3	Modell an Controller	335
5.1.4	Undo und Redo	337
5.1.5	Unittests	341
5.2	Als die Views das Laufen lernten	347
5.2.1	Blockfunktionen in C	349
5.2.2	Animationen mit Blöcken	353
5.2.3	Transitionen	357
5.2.4	Zur Animation? Bitte jeder nur einen Block!	358
5.3	Core Animation	360
5.3.1	Layer	360
5.3.2	Vordefinierte Layerklassen	366
5.3.3	Unser Button soll schöner werden	367
5.3.4	Spieglein, Spieglein an der Wand	369
5.3.5	Der bewegte Layer	372
5.3.6	Der View, der Layer, seine Animation und ihr Liebhaber	376
5.3.7	Die 3. Dimension	380
5.4	Der Tabbar-Controller	383
5.4.1	Aufbau einer Reiternavigation	383
5.4.2	Für ein paar Controller mehr	384
5.5	Was Sie schon immer über Instruments wissen wollten, aber nie zu fragen wagten	388
5.5.1	Spiel mir das Lied vom Leak	390
5.5.2	Ich folgte einem Zombie	394
5.5.3	Time Bandits	396

6	Programmieren, aber sicher	399
6.1	Sicherheitsmechanismen von iOS	400
6.2	Bedrohungen, Angriffe, Sicherheitslücken und Maßnahmen	403
6.2.1	Arten von Sicherheitslücken	405
6.3	Threat Modeling	406
6.3.1	Erstellen eines Datenflussdiagramms	408
6.3.2	STRIDE	414
6.3.3	Generische Design-Grundsätze	421
6.3.4	Threat Modeling aus der Tube – das Microsoft SDL Threat Modeling Tool	422
6.4	Sicherer Entwicklungszyklus	428
6.4.1	Awareness	430
6.4.2	Umgebung	432
6.4.3	Training	433
6.4.4	Dokumentation	434
6.4.5	Requirements	434
6.4.6	Design	436
6.4.7	Implementierung	437
6.4.8	Security-Testing	439
6.4.9	Deployment	440
6.4.10	Security Response	441
6.4.11	Sicherheitsmetriken	441
6.4.12	Abschließende Bemerkung	442
6.5	Sicherheit in der iOS-API	443
6.5.1	Keychain	443
6.5.2	Dateiattribute	451
6.5.3	Jailbreak-Erkennung	452
6.5.4	Event-Handling	456
7	Jahrmarkt der Nützlichkeiten	457
7.1	Xcode 4	457
7.1.1	Tabs	458
7.1.2	Organizer	460
7.2	Das Buildsystem	462
7.2.1	Workspaces, Projekte und Targets	462
7.2.2	Klassen in Bibliotheken auslagern	463
7.2.3	Bibliotheken wiederverwenden	469
7.2.4	Konfigurationen	472

Inhalt

7.2.5	Targets	476
7.2.6	Schemata	481
7.3	Ad-Hoc-Distributionen	485
7.3.1	Geräteregistrierung	486
7.3.2	Installation über das iPhone Konfigurationsprogramm	489
7.3.3	Ad-Hoc-Distributionen über einen Webserver	491
7.4	Versionsverwaltung mit Git	492
7.4.1	Git und Xcode	493
7.4.2	Git ohne Xcode	499
7.5	Abschließende Bemerkung	503
A Die Buch-DVD		505
Index		507

»Never put off till tomorrow what you can do the day after tomorrow.«
– Mark Twain

3 Aufwachen – analoger Wecker

Ab hier geht es ans Eingemachte. In diesem Kapitel werden Sie einen analogen Wecker programmieren. Dabei stehen die Arbeit mit Xcode und die praktische Anwendung der in den ersten beiden Kapiteln ausgeführten theoretischen Grundlagen im Vordergrund. Außerdem gestalten Sie eine einfache eigene Oberfläche. Denn das wichtigste Merkmal eines analogen Weckers ist schließlich das Ziffernblatt mit den Zeigern.

3.1 Eigene View-Klassen in Cocoa Touch

Und los geht's: Legen Sie ein neues Projekt in Xcode an, und geben Sie dem Projekt den Namen *AlarmClock*. Der Projekttyp basiert, analog zum *HelloWorld*-Beispiel aus Kapitel 2, auf einem View.

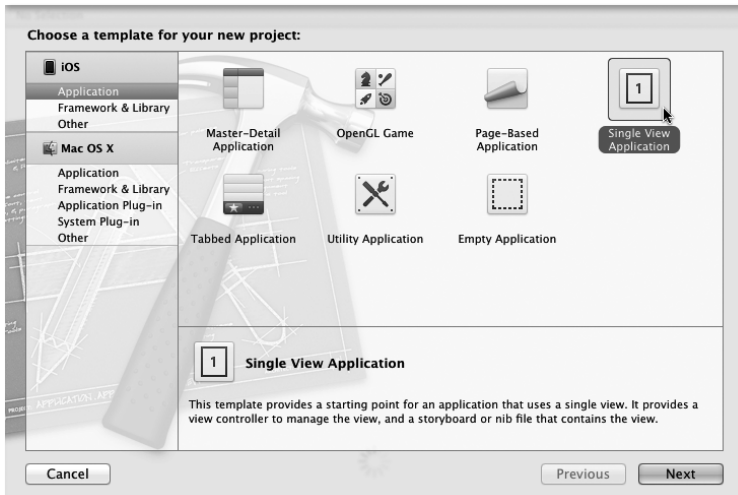


Abbildung 3.1 Auswahl der Projektvorlage für den Wecker

Im zweiten Schritt sollten Sie alle Checkboxen ausschalten. Als Produktnamen und Präfix für die Klassennamen geben Sie »AlarmClock« ein (siehe Abbildung 3.2).

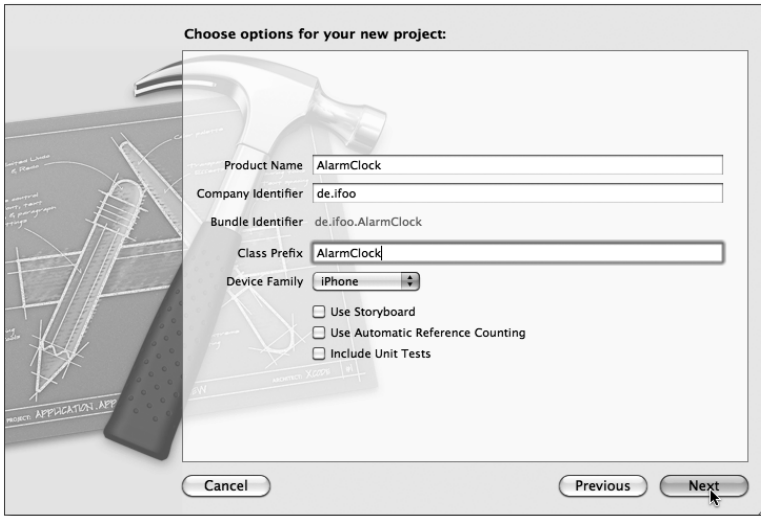


Abbildung 3.2 Die Parameter für das neue Projekt

Als Erstes muss der Wecker ein Ziffernblatt erhalten. Da es dafür keinen fertigen View in Cocoa Touch gibt, müssen Sie eine eigene View-Klasse erstellen. Zum Anlegen der Klasse klicken Sie mit der rechten Maustaste auf die Gruppe ALARM-CLOCK und wählen den Punkt NEW FILE... aus.

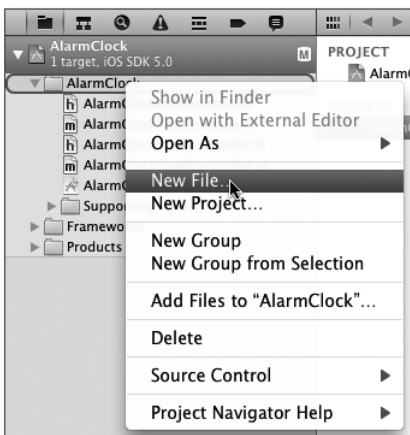


Abbildung 3.3 Hinzufügen einer neuen Klasse

Wählen Sie in der linken Spalte den Punkt COCOA TOUCH in der Rubrik IOS aus. Auf der rechten Seite erscheint eine Liste mit Dateivorlagen, aus der Sie OBJECTIVE-C CLASS auswählen. Geben Sie im nächsten Schritt der Klasse den Namen ClockView, und wählen Sie als Superklasse UIView aus (siehe Abbildung 3.4) – Xcode erstellt die Klassen dann automatisch als Subklasse von UIView.

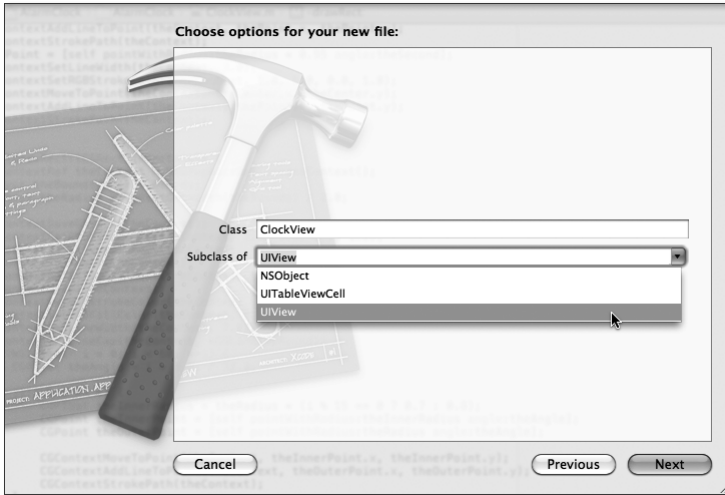


Abbildung 3.4 Subklasse von »UIView«

Im letzten Schritt wählen Sie den Ordner, die Gruppe und das Target für die Header- und die Implementierungsdatei aus. Hier können Sie einfach die Voreinstellungen übernehmen.

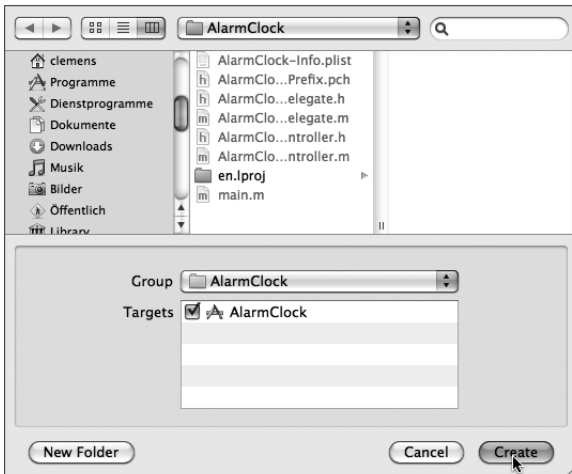


Abbildung 3.5 Anlegen der Klasse »ClockView« in der Gruppe »AlarmClock«

Nach dem Anlegen der Dateien offenbart ein Blick in den Quelltext der Header-datei der neuen Klasse, dass Xcode eine Subklasse von `UIView` erstellt hat:

```
#import <UIKit/UIKit.h>

@interface ClockView : UIView

@end
```

Listing 3.1 Header-Datei der Klasse »ClockView«

Um diese neue Viewklasse benutzen zu können, müssen Sie dafür zunächst einen View in der XIB-Datei des Viewcontrollers anlegen, die Xcode beim Anlegen des Projekts erzeugt hat. Ziehen Sie dazu im Interface Builder, den Sie durch Auswählen der Datei *AlarmClockViewController.xib* aktivieren, einen `UIView` aus der Objektbibliothek auf den bereits vorhandenen View. Dieser Subview wird später das Ziffernblatt in der Wecker-App darstellen.

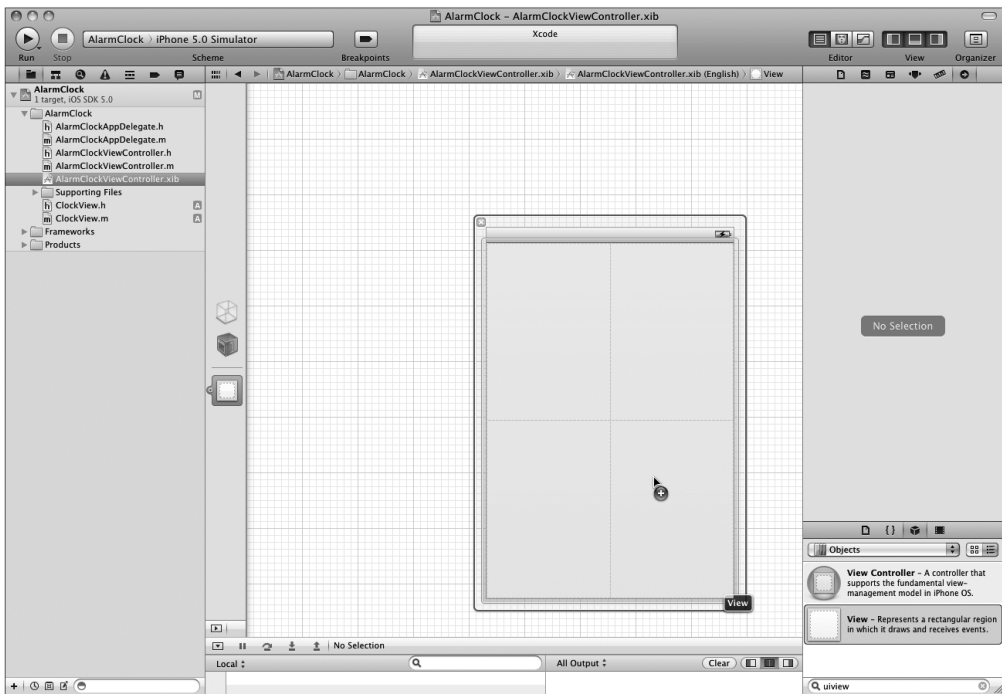


Abbildung 3.6 Ein neuer View im alten

Der neue View hat automatisch dieselbe Größe wie der bereits vorhandene erhalten. Da das Ziffernblatt der Weckers aber nicht den ganzen Bildschirm ausfüllen soll, markieren Sie den gerade hinzugefügten View und öffnen rechts oben im

Xcode-Fenster den Größen-Inspector (SHOW THE SIZE INSPECTOR in Abbildung 3.7 beziehungsweise $\text{ALT} + \text{CMD} + \text{5}$).

Setzen Sie die Koordinaten und die Größe des Views so, wie in Abbildung 3.7 dargestellt. Der Koordinatenursprung in einem View ist immer die linke obere Ecke. Die horizontale Ausdehnung wächst nach rechts und die vertikale nach unten.

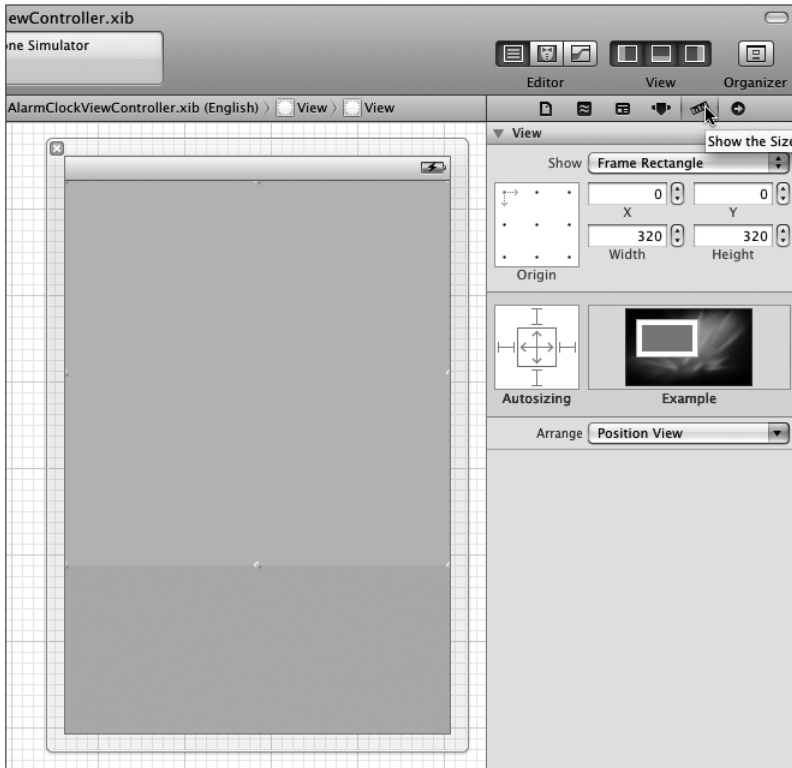


Abbildung 3.7 Der Größen-Inspector des Views

Das Ziffernblatt belegt mit dieser Einstellung nur circa zwei Drittel der oberen Fläche des iPhone-Bildschirms. Sie können das überprüfen, indem Sie das Projekt ausführen. Der Simulator zeigt jetzt eine weiße und darunter eine etwa halb so große graue Fläche an.

Über das dritte Symbol von links (neben dem Größen-Inspector) oder $\text{ALT} + \text{CMD} + \text{3}$ wählen Sie den Identitätsinspektor aus. Legen Sie dort über das oberste Eingabefeld, CLASS, die Klasse des Views fest. Sie können den Klassennamen dort entweder direkt eingeben oder aus einer Liste auswählen (siehe Abbildung 3.8). Weisen Sie dem View die neu angelegte Klasse `ClockView` zu.

Index

A

Abstrakte Klassen 52
Accessor 78
Action 168
 -*Methode* 176
Activityindicator 271
Address Book Framework 21
addTarget:action: 191
Ad-Hoc 485
afconvert (Kommandozeilenprogramm)
 203
Agil 428
aGPS 22
Akkulaufzeit 38
Alertbox 199
alpine 402
Analyzer 133, 216
Angemessenheit 435
Angriff 399, 403
Anonyme Kategorie 92
ANSI-C 53
Anzeigeausrichtung 251
App ID 228
App Store 17, 43
App-Icon 206
AppKit 24
Apple A5 34
AppleTV 19
Application-Delegate 96, 166, 196
applicationDidFinishLaunching
 withOptions 41
ARC 133
ARM 33
AS/NZS 406
ASLR 402
AT&T 18
Attack surface reduction 422
Attribut 54
 indiziertes 237
 optionales 237
 transientes 237
Attributinspektor 153
Audio-Toolbox 203
Authentisierung 414

Authorisierung 415
Automatic Reference Counting 133
 Speicherlecks und 390
 Zombies und 396
Autoreleasepool 74, 118
Autoresizingmask 214, 247
AVAudioPlayer 270
AVAudioRecorder 276
AVAudioSession 276
awakeFromNib 162
Awareness 430

B

Background 106
Backup 433
Badge 199, 384
Baumdarstellung 175
bedingte Kompilierung 319
Bedrohung 403
Bedrohungsanalyse 406
Bedrohungsdatenbank 437
beginTrackingWithTouch:withEvent: 187
Beispielprogramm
 AlarmClock 167
 Clock 167
 Games 325
Beispielprojekt
 Layer 366
 TableView 280
Benachrichtigung 107, 231, 294
 lokale 199
Benutzercode 444
Benutzerschnittstelle 39
Beobachtermuster 111
Berührungsgeste 179
Betriebsdokumentation 434
Betriebsumgebung 440
Bewegungssensor 30
Bibliotheken 437, 464
Bildgröße
 logische 208
 physikalische 208
Binäranalyse 439
Bitmaske 182, 247

Bitmenge 355
Blacklist 437
Block 349
Blockfunktion 349
Bluetooth 30
Bonjour 21
bounds 152
Breakpoint 217
BSD-Unix 18, 19
Buffer Overflow 56, 405
Bugtracking 432
Buildsystem 462
Bundle Identifier 95, 228
Bundle Seed ID 228

C

c't 399
CAAction 377
CAAnimation 372
CAAnimationGroup 375
CABasicAnimation 373
CAKeyframeAnimation 373
CALayer 360
cancelTrackingWithEvent: 187
CAPropertyAnimation 373
CAScrollLayer 367
CASHapeLayer 367
CATextLayer 367
CATransform3D 373, 380
CATransition 375
CDMA 34
Certificates 444
CFNetwork 21
CGContextFillPath() 153
CGContextRef 151
Cisco 20
class 55
Clean Desk Policy 432
Closure 349
Cocoa 21
Cocoa Touch 21, 23
Code-Analyse 439
Code-Kommentierung 438
Codesign 25
Code-Signierung 401
Company 26
Completionblock 359
Compliance 435

components:fromDate: 159
Concurrent Versions System 63
Containerview 174
Containerviewcontroller 114, 301, 321
contentView 281
Continue 457
continueTrackingWithTouch:withEvent:
187
Control 176
Controlzustand 182
Convenience-Konstruktor 120
convertPoint:fromView: 186
convertPoint:toView: 186
Core Animation 23, 360
Core Audio 23
Core Data 129, 234
Core Foundation 21, 22
Core Graphics 151
Core Location 21, 22
Core OS 21
Core Services 21
CPSSE 429
CPU 33
Crash 457
CVS 27
CVSSE 406

D

Dangling Pointer 115
Darwin 19
Dateiattribut 451
Dateisysteme 31, 460
Datenfluss 412
Datenflussdiagramm 407
Datenmodellinspektor 236
Datenquelle 293
Datensparsamkeit 422
Datenspeicher 412
Datenverbindungen 39
Debugger 80, 217
Debugging 460
Debug-Information 440
Defense in depth 422
Delegate 99, 103
delegate method 362
Denial of service 414, 420
Deployment 440
dequeueReusableCellWithIdentifier: 280

description 217
 Deserialisierung 98
 Design 436
 Designfehler 405
 Design-Grundsätze 421
 Designerter Initializer 77
 DetailviewController 309
 Developer Enterprise Program 223
 Developer Program 223
 didRotateFromInterfaceOrientation: 251
 DNS 21
 Dokumentation 434
 Doppel-Tap 181
 Douglas Adams 415

E

Ebene 1 409
 Eigentümer einer NIB-Datei 168
 Einfachvererbung 49
 Elevation of privileges 414, 421
 endTrackingWithTouch:withEvent: 187
 Entität 410
 Entitätstyp 235
 Entitlement 445
 Entwicklerprogramme 222
 Ereignis 179
 Value Changed 178
 Event 177
 Event-Handling 456
 EXC_BAD_ACCESS 222
 Extreme Programming 428

F

Fehler 401
 Fehlerbehandlung 438
 File's Owner 173
 filteredArrayUsingPredicate: 299
 Fingerposition 177
 frame 152
 Fremdcode 434, 435
 Fuzzing 439

G

Game Kit 24
 Garbage Collection 74
 Garbage Collector 74

GCC 53
 Generic Passwords 444
 genstrings (Kommandozeilenprogramm)
 211
 Geräteausrichtung 251
 Geräteregistrierung 486
 Geräteverwaltung 460
 Gesten 23
 Gesturerecognizer 189
 Getter 47
 Git 27, 63, 492
 GitX 499, 505
 Go-Live-Test 441
 GPRS 33
 GPS 22
 Grafikkontext 151
 GSM 33

H

Hardware 32
 Hardware-Tastatur 254
 Headerdatei 67
 Heap 56, 115, 402
 Heise 399
 Hintergrundfarbe 153
 hitTest:withEvent: 186
 Hochformat 246
 Home-Taste 42
 HSDPA 33

I

IBAction 176
 IBOutlet 170
 IBOutletCollection 172
 iControl 400
 id 55
 imageNamed: 208
 Implementierung 437
 Implementierungsfehler 405
 Implementierungsrichtlinie 438
 Inaktiv 106
 Individual 26
 Info.plist 212
 InfoPlist.strings 212
 Information disclosure 414, 419, 420
 Initializer 71
 initWithCoder: 162

initWithNibName:bundle: 198
 Input-Accessory-View 255
 inputAccessoryView 258
 Installationshandbuch 440
 Installationsprozess 440
 Instanzvariable 54
 Instruments 29, 388
 Integrität 414
 Integritätsbedingungen 243
 interfaceOrientation 251
 Internet Passwords 444
 iOS 17, 18
 Dev Center 25
 Developer Program 26
 Developer University Program 27
 Development Guide 31
 Enterprise Program 26
 iPad 18, 34
 iPhone 17
 3G 33
 3GS 33
 4 33
 iPhone OS 20
 iphone-dataprotection 446
 iPod touch 18
 ISMS 431
 ISO 27001 431
 ISSECO 429
 Issue-Navigator 215
 isViewLoaded 198
 ivar 54

J

Jailbreak 32, 401

K

Kamera 30
 Kapselung 46
 Kategorie, anonyme 164
 kCFBooleanTrue 451
 Keep it simple, stupid 422
 Kennung für Tabellenzellen 280
 Kennzahlen 439
 Keychain 403, 443
 KeychainViewer 447
 Keypath 130
 Keys 444

Key-Value-Coding 129
 Key-Value-Observing 336
 Klasse 47
 Klassenmethode 60
 Konfiguration 472
 Konsistenz 326, 327
 Konstruktor 71
 Kontextdiagramm 408
 Kreditkarte 401
 Kryptografie 437
 kSecAttrAccessibleAfterFirstUnlock 445
 kSecAttrAccessibleAfterFirstUnlock-
 ThisDeviceOnly 445
 kSecAttrAccessibleAlways 445
 kSecAttrAccessibleAlwaysThisDevice-
 Only 445
 kSecAttrAccessibleWhenUnlocked 445
 kSecAttrAccessibleWhenUnlocked-
 ThisDeviceOnly 445
 kSecReturnAttributes 451
 KVC 129
 KVO 336

L

Last-In-First-Out 338
 Laufzeitfehler 215, 217
 Layer 354
 layerClass 360
 layoutSubviews 252
 Lazy-Getter 205
 Leak 115
 Least privilege 422
 LibSystem 21
 LIFO 338
 Linker 25
 Linux 401
 loadNibNamed:owner:options: 281
 loadView 198
 Localizable.strings 211
 Location Suite 505
 locationInView: 191
 Logging 216
 Lokalisierung 30, 210
 Lose Kopplung 111

M

Mac App Store 25
 Mac OS X 18
 Mach 19
 MainWindow.nib 173
 MainWindow.xib 195, 213
 MainWindow-iPad.xib 214
 Map Kit 24
 Maßnahme 403
 Masterviewcontroller 309
 Media 23
 Mehr-Eintrag 384
 Mehrfachvererbung 49
 Message Passing 46
 Message UI Framework 24
 Messgröße 442
 Microsoft 407
 SDL Pro Network 422
 SDL Threat Modeling Tool 422
 Threat Analysis & Modeling 427
 Mikrokern 19
 mobile 402
 Mobilfunk 30
 Modultest 341
 Multithreading 137
 Multitouch 33
 mutableCopy 301
 Mutator 47
 MVC-Architekturmuster 35, 187

N

Nachrichten 52
 Nachrichtenaustausch 52
 Navigationcontroller 302
 Navigationitem 304
 Navigationsleiste 301, 304
 Need to know 422
 NeXT 19
 NEXTSTEP 19
 NIST 408
 Non repudiation 414
 Notification 231
 NSCalendar 159
 NSDate 159, 201
 NSDateFormatter 159
 NSDictionary 451
 NSEntityDescription 242

NSFetchedResultsController 292
 NSFetchedResultsControllerDelegate 294
 NSFetchRequest 287
 NSFileProtectionComplete 451
 NSFileProtectionCompleteUnlessOpen 452
 NSFileProtectionCompleteUntilFirstUserAuthentication 452
 NSFileProtectionKey 451, 452
 NSFileProtectionNone 451
 NSIndexPath 280
 NSInvocation 337
 NSKeyValueCoding 129
 NSLocalizedString 211
 NSLog 216, 460
 NSManagedObjectContext 242
 NSMutableArray 301
 NSMutableDictionary 449
 NSNotification 231
 NSNotificationCenter 232
 NSNumber 206
 NSPredicate 289
 NSSet 238
 NSSortDescriptor 288
 NSString 55
 NSUndoManager 337
 numberOfSectionsInTableView: 279
 NX 402

O

Objective-C 19, 53
 Objekte 46, 47
 Objektgraph 235
 Objektkontext 242
 Objektmethode 60
 Objektorientierte Programmierung 44
 Objektserialisierung 98
 OCTAVE 407
 OpenAL 23
 OpenGL ES 23
 OpenSSL 22
 Organizer 216, 460
 orientation 251
 Orthogonal 189
 Outlet 168, 198
 -*Attribut* 170, 173
 -*Collection* 169, 172
 -*Property* 170

P

PageviewController 310, 314
 Paketbuilder 25
 PAL 415
 PDCA 431
 Pentesting 439
 Perforce 27
 Performanceleak 398
 performSegueWithIdentifier:sender: 193
 Persistenz 235
 pointInside:withEvent: 186, 188
 Polymorphie 46
 Popovercontroller 194
 Popoverviewcontroller 301, 310
 popViewControllerAnimated: 302
 POSIX 21
 PowerPC 25
 Prädikat 288
 Präsentationslayer 370
 prepareForSegue:sender: 194
 Project Builder 21, 457
 Projekte 462
 Property
 Read-Only-Eigenschaft 165
 Protection class 445
 Provisioning Profile 32
 Proxy 340
 Proxymuster 340
 Prozess 411, 429
 Prozessgruppe 411
 pushViewController:animated: 302

Q

Qmail 442
 Quelle 410
 Querformat 246
 Quicktime 23

R

Redo 242
 Redo-Stack 338
 Reference Counter 116
 Referenzzahlen 115, 116
 registerNib:forCellReuseIdentifier: 285
 Reiternavigation 383

Relationships 235
 Many-To-Many 238
 Many-To-One 238
 One-To-Many 237
 removeTarget:action: 191
 Repudiation 414, 419
 Requirement 434
 Response-Policy 441
 Ressourcenverzeichnis 277
 Retina Display 33, 208
 Rhapsody 19
 Richtlinien 436
 root 402
 RootviewController 302
 Rotation an den Hauptachsen 382
 RUP 428

S

Sandbox 42, 277
 Sandboxing 401
 Schadsoftware 453
 Schemata 481
 Schlüsselbund 403
 Schlüsselmanagement 440
 Schwache Referenz 135
 Schwachstellenanalyse 439
 SCM 27, 461
 Scrum 428
 SDK 25
 SDL 429
 SecItemAdd 449
 SecItemCopyMatching 451
 SecItemDelete 449
 Secure by default 422
 Secure Coding Guide 400
 Security Advisor 435
 Security Framework 21
 Security Policy 430
 Security Response 441
 Security-Datenbank 434
 Security-Expertise 435
 Security-Framework 448
 Security-Testing 439
 Security-Training 433
 Segmented Control 251, 356
 Segue 169, 193, 321
 Selektor 56
 Sendmail 442

Senke 410
 Serialisierung 162
 setNeedsDisplay 152
 Setter 47
 setuid 402
 Sicherer Entwicklungszyklus 428
 Sicherheit 42
 Sicherheit der Ausstattung 432
 Sicherheitsanforderungen 435
 Sicherheitskonzept 434
 Sicherheitslücke 403
 Sicherheitsmetrik 441
 Sicherheitsüberprüfung 435
 SIGART 221
 Simula 45
 Simulator 30
 Singleton 105, 138
 Skalierungsfaktor 208
 Smalltalk-80 46
 Snapshot 142, 461
 Snow Leopard 25
 Sockets 21
 Software-Tastatur 254
 Source-Code-Managementsystem 63
 Spaghetticode 111
 Speicher 37
 Speicherleck 390
 Speicherverwaltungsregeln
 und Outlet-Attribute 170
 Speicherwarnungen 37
 Spiralmodell 428
 SplitviewController 309
 Spoofing 414, 417
 SQLite 21
 Stack 56, 402
 Stacktrace 221
 Standort 36
 Storyboard 95, 193, 196, 284
 strcpy 405
 STRIDE 414
 Subklasse 48
 sublayerTransform 381
 Subversion 27
 SubviewController 321
 Sun Tzu 404
 Superklasse 48
 sys_setreguid 402
 sys_setreuid 402

System Tools 461
 systemVersion 319

T

Tabbarnavigation 383
 Tableview 278
 tableView:cellForRowAtIndexPath: 280
 tableView:heightForRowAtIndexPath:
 284
 tableView:numberOfRowsInSection: 279
 Tabs 458
 Tag 283
 Tampering 414, 418
 Target 342, 462
 Target-Action 176
 -Mechanismus 187
 Targets 462, 476
 Tastatur 254
 Team Provisioning Profile 227
 temporäre Datei 277
 Test 31
 Testdaten 433
 Testmethode 345
 Test-Methodik 439
 Testobjekt 345
 textViewDidEndEditing: 254
 textViewShouldBeginEditing: 254
 Threat Modeling 406, 433
 Time Profiler 398
 toolbarItems 307
 Tools 434
 touchesBegan:withEvent: 187
 touchesCancelled:withEvent: 187
 touchesEnded:withEvent: 187
 touchesMoved:withEvent: 187
 Touchpad 32
 Training 433
 Trainingsplan 433
 Triangulation 36
 Trust boundary 413
 Trust-Modelle 436
 TU Wien 402
 Typlosigkeit 55

U

Übergang 193
 UDID 402

- UIAlertView 202
- UIApplication 174, 196, 200
- UIApplicationMain 41
- UIBarButtonItem 305
- UIControl 176
- UIControlEventEditingDidBegin 181
- UIControlEventEditingDidChange 181
- UIControlEventEditingDidEnd 182
- UIControlEventEditingDidEndOnExit 182
- UIControlEventTouchCancel 180
- UIControlEventTouchDown 180
- UIControlEventTouchDownRepeat 180
- UIControlEventTouchDragEnter 180
- UIControlEventTouchDragExit 180
- UIControlEventTouchDragInside 180
- UIControlEventTouchDragOutside 180
- UIControlEventTouchUpInside 180
- UIControlEventTouchUpOutside 180
- UIControlEventValueChanged 179
- UIControlStateDisabled 182
- UIControlStateHighlighted 182
- UIControlStateSelected 182
- UIDevice 251, 319
- UIGestureRecognizer 189
- UIImage 208
- UIKit 23
- UILocalNotification 200
- UILongPressGestureRecognizer 190
- UINavigationController 304
- UINavigationController 301
- UIPageViewController 313
- UIPageViewControllerDataSource 316
- UIPageViewControllerDelegate 317
- UIPanGestureRecognizer 190
- UIPinchGestureRecognizer 190
- UIPopoverController 194
- UIResponder 186
- UIRotationGestureRecognizer 190
- UIScreen 210
- UISearchBar 298
- UISearchDisplayController 298
- UISplitViewController 309
- UISwipeGestureRecognizer 190
- UISwitch 177
- UITabBarController 383
- UITabBarItem 384
- UITableView 279
- UITableViewCell 280
- UITableViewDataSource 279
- UITableViewDelegate 279
- UITapGestureRecognizer 190
- UITextField 254
- UITextView 254
- UIToolBar 258
- UITouch 177
- UIView 186
- UIViewController 168, 186
- UIWebView 319
- Umgebung 432
- UMTS 33
- Unabhängige Viewklasse 166
- Undo 242
- Undo-Stack 338
- Unittest 341
- Unix 18
- Unix V1 18
- URL 24
- userInteractionEnabled 185

V

- Vektor 381
- Verbindungsinspektor 170
- Vererbung 46, 47
- Verfügbarkeit 415
- Vergleichsoperator 290
- Verizon 34
- Verschlüsselung 460
- Versionsverwaltung 433, 492
- Vertraulichkeit 415
- view 198
- ViewController 168
 - Anzeigezyklus 199
 - Ladezyklus 199
 - Lebenszyklus 197
- viewDidAppear: 168, 199
- viewDidDisappear: 199
- viewDidLayoutSubviews 324
- viewDidLoad 198
- viewDidUnload 173
- Viewhierarchie 175
- viewWillAppear: 199
- viewWillDisappear: 168, 199
- viewWithTag: 283
- V-Modell 428
- Vorabkontrolle 435

W

Werkzeugleiste 257, 307
Wildcard-Profilen 227
willRotateToInterfaceOrientation:duration: 251
Wischnavigation 313
WLAN 33
Workspaces 462
WPS 22

X

Xcode 27, 457
 Version 4.2 190

Z

Zeitverbrauch 396
Zellprototyp 284
Zertifikat 32
Zertifikatsanfrage 32
Zombie 394
zPosition 380
Zurück-Button 306