

Indice generale

Parte I	Per cominciare.....	1
Capitolo 1	Il libro e voi	3
	C e Objective-C.....	4
	Struttura del libro	4
	La vita di un programmatore.....	5
Capitolo 2	Il vostro primo programma.....	7
	Installare gli strumenti per sviluppatori di Apple.....	7
	Primi passi con Xcode.....	8
	Da dove si comincia per scrivere il codice?.....	10
	Come si esegue il programma?.....	14
	Ma che cos'è un programma?	15
	Non fermatevi mai	17
Parte II	La programmazione	19
Capitolo 3	Variabili e tipi	21
	I tipi delle variabili.....	22
	Un programma con variabili	23
	La sfida.....	24
Capitolo 4	if/else	27
	Variabili booleane	29
	Quando le parentesi graffe sono opzionali.....	29
	else if.....	30
	Per i più curiosi: gli operatori condizionali	31
	La sfida.....	31

Capitolo 5	Le funzioni.....	33
	Quando è il caso di utilizzare una funzione?.....	33
	Come si scrive una funzione e come si usa?	34
	Le funzioni possono lavorare insieme	36
	Le librerie standard	37
	Le variabili locali, i frame e lo stack.....	38
	Ambito.....	40
	Ricorsione	41
	Esaminare i frame nel debugger	43
	L'istruzione return	45
	Variabili globali e statiche.....	47
	La sfida.....	48
Capitolo 6	Stringhe di formato	49
	I token	50
	Sequenze di escape	50
	La sfida.....	51
Capitolo 7	I numeri.....	53
	I numeri interi	53
	Token per visualizzare interi.....	54
	Operazioni sugli interi.....	56
	Numeri a virgola mobile	58
	Token per visualizzare numeri a virgola mobile	59
	La libreria math.....	59
	La sfida.....	60
	Nota sui commenti.....	60
Capitolo 8	I cicli	61
	Il ciclo while	62
	Il ciclo for	63
	break	64
	continue.....	65
	Il ciclo do-while.....	66
	La sfida: conto alla rovescia.....	67
	La sfida: input dell'utente	68
Capitolo 9	Indirizzi e puntatori	71
	Come ottenere gli indirizzi.....	72
	Memorizzazione di indirizzi in puntatori.....	73
	Come ottenere i dati a un determinato indirizzo.....	73
	Quanti byte?.....	74
	NULL.....	75
	Dichiarazioni eleganti per i puntatori.....	76
	La sfida: quanta memoria?.....	77
	La sfida: quale intervallo?	77

Capitolo 10	Passaggio per riferimento.....	79
	Scrivere funzioni con passaggio per riferimento	80
	Evitare di dereferenziare NULL	82
	La sfida.....	82
Capitolo 11	Le strutture.....	83
	La sfida.....	86
Capitolo 12	Lo heap	87
Parte III	Objective-C e il framework Foundation.....	91
Capitolo 13	Gli oggetti.....	93
	Che cosa sono gli oggetti.....	93
	Le classi	94
	Creare il primo oggetto	95
	Metodi e messaggi.....	97
	Inviare messaggi.....	97
	Un altro messaggio	98
	Metodi di classe e metodi istanza a confronto	98
	Inviare messaggi sbagliati.....	99
	Nota sulla terminologia.....	102
	La sfida.....	102
Capitolo 14	Ancora messaggi.....	105
	Un messaggio con un argomento.....	105
	Argomenti multipli.....	106
	Invii di messaggi annidati.....	108
	alloc e init	109
	Invio di messaggi a nil	110
	id	111
	La sfida.....	111
Capitolo 15	Oggetti e memoria.....	113
	Puntatori e loro valori	113
	Gestione della memoria.....	116
	ARC	117
Capitolo 16	NSString.....	121
	Creare istanze di NSString.....	121
	Metodi di NSString.....	122
	Guide di riferimento per le classi	123
	Altre parti della documentazione	127
	La sfida: trovare altri metodi di NSString.....	128
	La sfida: usare readline()	128

Capitolo 17	NSArray.....	129
	Creazione di array	129
	Accesso agli array.....	130
	Iterare sugli array	133
	NSMutableArray	135
	Metodi vecchio stile per gli array	136
	La sfida: una lista della spesa	137
	La sfida: nomi interessanti	137
Capitolo 18	La vostra prima classe	139
	Metodi di accesso	143
	Convenzioni per i nomi dei metodi di accesso	144
	self	145
	File multipli.....	145
	Prefissi di classe.....	146
	La sfida	146
Capitolo 19	Le proprietà.....	147
	Dichiarazione di proprietà	147
	Attributi di proprietà	149
	La notazione puntata	150
Capitolo 20	L'ereditarietà	153
	Ridefinizione di metodi	157
	super	158
	Gerarchia di ereditarietà.....	158
	description e %@	160
	La sfida	161
Capitolo 21	Oggetti variabili istanza e proprietà.....	163
	Proprietari di un oggetto e ARC	166
	Creazione della classe BNRAsset	166
	Aggiunta di una relazione “a-molti” a BNREmployee.....	167
	La sfida: portafoglio azionario	171
	La sfida: rimozione di asset.....	172
Capitolo 22	Estensioni di classe	173
	Nascondere la mutabilità.....	175
	Header ed ereditarietà	176
	Header e variabili istanza generate	176
	La sfida	176

Capitolo 23	Come evitare i leak in memoria	177
	Riferimenti circolari forti	180
	Riferimenti deboli.....	182
	Annullamento dei riferimenti deboli.....	183
	Per i più curiosi: conteggio manuale dei riferimenti e storia di ARC ..	185
	Regole per i retain count.....	186
Capitolo 24	Classi collezione	189
	NSSet/NSMutableSet	189
	NSDictionary/NSMutableDictionary	192
	Oggetti immutabili	195
	Ordinamento di array	196
	Filtraggio.....	198
	Collezioni e proprietà.....	199
	Tipi primitivi del linguaggio C.....	199
	Collezioni e nil.....	200
	La sfida: approfondire.....	200
	La sfida: pacchetti azionari di maggior valore.....	200
	La sfida: pacchetti azionari ordinati	201
Capitolo 25	Le costanti	203
	Direttive del preprocessore.....	204
	#include e #import.....	205
	#define.....	206
	Variabili globali.....	206
	enum.....	208
	#define e variabili globali a confronto.....	209
Capitolo 26	Scrittura di file con NSString e NSData	211
	Scrivere una NSString su un file	211
	NSError	212
	Lettura di file con NSString.....	214
	Scrittura di un oggetto NSData in un file.....	215
	Lettura di un NSData da un file.....	216
	Trovare directory speciali	217
Capitolo 27	I callback.....	219
	Il ciclo di esecuzione	220
	Target-azione	221
	Oggetti ausiliari	223
	Notifiche.....	227
	Quale tipo di callback usare?	228
	Callback e proprietà dell'oggetto	228
	Per i più curiosi: come funzionano i selettori	229

Capitolo 28	I blocchi.....	233
	Uso dei blocchi	234
	Dichiarazione di una variabile blocco	234
	Composizione di un blocco	235
	Passaggio di un blocco	237
	typedef	238
	Blocchi e altri callback a confronto	239
	Ancora sui blocchi.....	240
	Valori di ritorno	240
	Blocchi anonimi	240
	Variabili esterne	241
	La sfida: un blocco anonimo	243
	La sfida: usare un blocco con NSNotificationCenter	244
Capitolo 29	I protocolli	245
	Chiamata di metodi opzionali.....	247
Capitolo 30	Liste di proprietà.....	249
	La sfida.....	252
Parte IV	Applicazioni guidate dagli eventi	253
Capitolo 31	La vostra prima applicazione iOS	255
	Applicazioni basate su GUI.....	255
	Primi passi con l'applicazione iTahDoodle	256
	BNRAppDelegate	258
	MVC	259
	Il delegato dell'applicazione	260
	Impostazione delle viste.....	262
	Esecuzione sul simulatore iOS	264
	Cablaggio del pulsante.....	266
	Cablaggio della vista tabella	268
	Salvataggio e caricamento dei dati.....	271
	Aggiunta di una funzione C di supporto	272
	Salvataggio dei dati delle attività.....	273
	Caricamento dei dati delle attività	274
	Per i più curiosi: che cosa dire di main()?	274
	Per i più curiosi: esecuzione di iTahDoodle su un dispositivo mobile	275

Capitolo 32	La vostra prima applicazione Cocoa.....	277
	Primi passi con TahDoodle	279
	Impostazione delle viste in Interface Builder	281
	Impostazione del pulsante	282
	Impostazione della vista tabella.....	284
	Aggiunta di vincoli di autolayout	286
	Creare connessioni	289
	Il proprietario del file.....	290
	Impostazione della coppia target-azione del pulsante	290
	Connessione della vista tabella.....	292
	Implementazione di NSTableViewDataSource	294
	Salvataggio e caricamento di dati	296
	La sfida.....	297
Parte V	Objective-C avanzato	299
Capitolo 33	init	301
	Scrivere metodi init	301
	Il più semplice metodo init.....	302
	instancetype.....	303
	Uso e verifica dell'inizializzatore di superclasse	304
	Metodi init che richiedono argomenti	304
	Uso dei metodi d'accesso.....	306
	Inizializzatori multipli.....	307
	Metodi init mortali.....	310
Capitolo 34	Ancora sulle proprietà	313
	Ancora sugli attributi delle proprietà	313
	Mutabilità.....	313
	Specificatori della durata di vita.....	313
	Un consiglio su atomic e nonatomic	316
	Implementazione di metodi di accesso	317
Capitolo 35	La codifica chiave/valore	319
	Tipi non oggetto	321
	Percorsi chiave.....	322
Capitolo 36	L'osservazione chiave-valore	325
	Uso del contesto nel KVO.....	326
	Generazione esplicita della notifica	327
	Proprietà dipendenti.....	328
Capitolo 37	Le categorie	329
	La sfida.....	330

Parte VI	C avanzato	333
Capitolo 38	Operazioni bitwise.....	335
	OR bitwise	336
	AND bitwise.....	338
	Altri operatori bitwise	339
	OR esclusivo	339
	Complemento	340
	Shift a sinistra (left-shift).....	341
	Shift a destra (right-shift).....	342
	Uso di enum per definire delle bitmask.....	342
	Byte e ancora byte	343
	La sfida	343
Capitolo 39	Le stringhe del linguaggio C	345
	char	345
	char *.....	346
	Letterali stringa.....	349
	Conversione in e da NSString	351
	La sfida.....	351
Capitolo 40	Gli array del linguaggio C.....	353
	La sfida	356
Capitolo 41	Esecuzione dalla riga di comando	361
	Argomenti della riga di comando.....	362
	Per un'esecuzione più comoda dalla riga di comando.....	365
Capitolo 42	L'istruzione switch	367
Appendice	Il sistema di runtime di Objective-C	369
	Introspezione.....	370
	Ricerca ed esecuzione dinamica di metodi.....	370
	Gestione di classi e gerarchie ereditarie	371
	Il meccanismo KVO	374
	Note conclusive.....	376
	La sfida: variabili istanza	376
I prossimi passi		377
	Contatti.....	378
Indice analitico.....		379

Il libro e voi

Parliamo un attimo di voi. Volete scrivere applicazioni per iOS o OS X, ma avete poca (o nessuna) esperienza di programmazione. I vostri amici vi hanno parlato con entusiasmo di altri libri, ma sono testi che si rivolgono a programmatori esperti. E allora che fare?

Leggete questo libro.

Esistono altri libri simili, ma voi dovete leggere proprio questo. E perché? Perché, grazie alla nostra lunga esperienza nell'insegnare come scrivere applicazioni per iOS e Mac, abbiamo individuato con precisione ciò che vi serve conoscere a questo punto del vostro percorso. Abbiamo lavorato duramente per distillare le conoscenze necessarie e scartare tutto il resto. In questo libro troverete molta sapienza e pochissima fuffa.

Il nostro è un approccio inconsueto. Aniché cercare semplicemente di farvi comprendere la sintassi di Objective-C, vi mostreremo come funziona la programmazione e come la vedono i programmatori esperti. A causa di questo approccio, in questo libro tratteremo da subito alcuni concetti difficili. Non aspettatevi una lettura facile. Inoltre, praticamente ogni concetto è accompagnato da un esperimento di programmazione. Questa combinazione che prevede l'apprendimento di concetti e la loro immediata applicazione pratica è il modo migliore per imparare a programmare.

In questo capitolo

- **C e Objective-C**
- **Struttura del libro**
- **La vita di un programmatore**

C e Objective-C

Quando eseguite un programma, un file viene copiato dal file system nella memoria (RAM) e le istruzioni in esso contenute vengono eseguite dal computer. Tali istruzioni sono illeggibili per l'uomo, che perciò scrive i programmi per computer utilizzando un linguaggio di programmazione. Il linguaggio di programmazione di linguaggio più basso (più vicino alla macchina) si chiama *assembly* e prevede che si descriva ogni passaggio che la CPU (il cervello del computer) deve effettuare. Il codice assembly viene poi trasformato in *codice macchina* (la lingua madre del computer) tra un *assembler*. Il linguaggio assembly è noiosamente prolisso e dipendente dalla CPU (perché il cervello del vostro nuovo iMac può essere molto diverso da quello di un vecchio e amato PowerBook). In altre parole, se volete eseguire il programma su un tipo diverso di computer, dovrete riscrivere il codice assembly.

Per ottenere codice che potesse essere facilmente portato da un tipo di computer a un altro sono stati sviluppati i “linguaggi di alto livello”. Con questi linguaggi, anziché dover pensare a un particolare tipo di CPU, potete esprimere le istruzioni in un modo più generale, e un programma apposito (detto *compilatore*) tradurrà il vostro codice in un codice macchina specifico della CPU e altamente ottimizzato. Un linguaggio di alto livello è il C. I programmatori in C scrivono codice nel linguaggio C, e un compilatore C traduce il codice C in codice macchina.

Il linguaggio C è stato creato all'inizio degli anni Settanta presso AT&T. Il sistema operativo Unix, che è ancora oggi alla base di OS X e Linux, è stato scritto in C con un po' di codice assembly per alcune operazioni di basso livello. Anche il sistema operativo Windows è per buona parte scritto in C.

Il linguaggio di programmazione Objective-C è basato sul linguaggio C, ma aggiunge il supporto per la programmazione orientata agli oggetti. È il linguaggio utilizzato per scrivere applicazioni destinate ai sistemi operativi iOS e OS X di Apple.

Struttura del libro

In questo libro imparerete i linguaggi di programmazione C e Objective-C a un livello sufficiente per sviluppare applicazioni destinate a Mac o dispositivi iOS.

Perché ci occupiamo prima del C? Perché ogni programmatore in Objective-C ha bisogno di una conoscenza abbastanza approfondita del linguaggio C. Inoltre, molti concetti che sembrano complicati in Objective-C hanno radici molto semplici in C. Spesso introdurremo un concetto usando il linguaggio C e poi vedremo come padroneggiarlo in Objective-C.

Questo libro è stato progettato per essere letto avendo di fronte un Mac. Troverete le spiegazioni dei concetti ed esperimenti pratici per illustrarli in concreto. Questi esperimenti non sono da considerarsi facoltativi: non capirete veramente il libro se non li svolgerete. Il modo migliore per imparare a programmare è quello di digitare il codice, commettere errori, risolverli e acquisire familiarità anche a livello fisico

con gli schemi del linguaggio. Leggere il codice e capire i concetti teorici non è sufficiente per acquisire le competenze necessarie.

Per fare ancora più pratica, al termine di ciascun capitolo sono riportati degli esercizi denominati “La sfida”, che vi consentiranno di acquisire ancora più familiarità con quanto avete appreso. Sugeriamo caldamente di affrontare tutte le “sfide” che potete. In alcuni capitoli, nelle pagine finali, ci sono anche paragrafi intitolati “Per i più curiosi”. Si tratta di spiegazioni più approfondite di argomenti trattati nel capitolo; non sono considerate assolutamente essenziali, ma speriamo che le troverete interessanti e utili.

L'editore americano Big Nerd Ranch ospita un forum dove i lettori discutono del libro e degli esercizi; potete trovarlo presso <http://forums.bignerdranch.com>.

Questo libro e la programmazione in generale vi risulteranno più piacevoli se siete capaci di utilizzare gli schermi sensibili al tocco; in questo modo, a parte la maggiore velocità, potrete osservare lo schermo e il libro anziché farvi distrarre dalla tastiera. Così è molto più facile rilevare gli errori di battitura non appena vengono commessi. Sono disponibili numerosi programmi per Mac che insegnano a utilizzare al meglio gli schermi touchscreen.

La vita di un programmatore

Acquistando questo libro avete preso la decisione di diventare programmatori. Dovete sapere di che cosa si tratta.

La vita di un programmatore è in gran parte una battaglia senza fine. La necessità di risolvere problemi all'interno di un panorama tecnologico in continuo mutamento obbliga i programmatori ad apprendere sempre nuove cose, o meglio a “combattere contro la propria stessa ignoranza”. Anche se un programmatore vuole soltanto risolvere un bug del codice che utilizza una tecnologia familiare, a volte il software che creiamo è talmente complesso che soltanto il compito di individuare il difetto può richiedere una giornata intera.

Se scrivete codice, dovete lottare. La maggior parte dei programmatori professionisti impara a lottare ora dopo ora, giorno dopo giorno, senza un'eccessiva frustrazione. Questa è un'altra competenza molto utile. Se siete curiosi di conoscere la vita dei programmatori e i moderni progetti software, vi consigliamo la lettura del libro *Dreaming in Code* di Scott Rosenberg.

E ora è giunto il momento di scrivere finalmente il primo programma.