

Stručný obsah

Úvod	13
------------	----

Část I

Jazyk Objective-C 2.0

1. Programování v Objective-C	21
2. Třídy, objekty a metody.....	39
3. Datové typy a výrazy	59
4. Programové smyčky	85
5. Rozhodování	105
6. Třídy zblízka	135
7. Dědičnost	157
8. Polymorfismus, dynamické typování a dynamická vazba	185
9. Více o proměnných a datových typech.....	201
10. Kategorie a protokoly	219
11. Preprocesor.....	233
12. Další prvky jazyka C.....	247

Část II

Základy frameworku Foundation

13. Úvod do frameworku Foundation.....	303
14. Čísla, řetězce a kontejnery.....	307
15. Práce se soubory.....	359
16. Správa paměti.....	383
17. Kopírování objektů	401
18. Archivování.....	413

Část III**Cocoa a SDK iPhone**

19. Úvod do Cocoa.....	433
20. Psaní aplikací pro iPhone.....	437

Část IV**Přílohy**

A. Glosář	469
B. Souhrn jazyka Objective-C 2.0	475
C. Zdrojový kód adresáře.....	529
D. Prostředky	535

Obsah

Úvod	13
-------------------	-----------

Část I

Jazyk Objective-C 2.0

Lekce 1

Programování v Objective-C	21
---	-----------

Překládání a spuštění programů	22
Použití Xcode	22
Použití aplikace Terminal	28
Vysvětlení prvního programu	30
Zobrazení hodnot proměnných	33
Souhrn	35
Cvičení.....	36

Lekce 2

Třídy, objekty a metody	39
--------------------------------------	-----------

Co je to vlastně objekt?	40
Instance a metody	40
Třída Objective-C pro práci se zlomky	42
Oddíl @interface.....	45
Výběr názvů.....	45
Instanční proměnné	47
Metody třídy a instancí.....	47
Oddíl @implementation.....	49
Oddíl vlastního programu	50
Přístup k instančním proměnným a zapouzdření dat	55
Souhrn	58
Cvičení.....	58

Lekce 3

Datové typy a výrazy..... 59

Datové typy a konstanty	60
Typ int	60
Typ float	61
Type double.....	62
Typ char.....	62
Kvalifikátory: long, long long, short, unsigned a signed.....	63
Typ id.....	65
Aritmetické výrazy.....	66
Priorita operátorů.....	66
Celočíselná aritmetika a unární operátor minus.....	68
Operátor modulo.....	70
Převody celých čísel a čísel s pohyblivou desetinnou čárkou.....	71
Operátor převodu typů	73
Operátory přiřazení.....	73
Třída Kalkulacka.....	74
Bitové operátory	76
Bitový operátor AND.....	77
Operátor bitového inkluzivního OR.....	78
Operátor bitového exkluzivního OR.....	78
Operátor komplementu.....	79
Operátor posunu doleva.....	80
Operátor posunu doprava.....	81
Typy _Bool, _Complex a _Imaginary	82
Cvičení.....	82

Lekce 4

Programové smyčky..... 85

Příkaz for	87
Zadání z klávesnice.....	93
Vnořené smyčky for	94
Varianty smyček for.....	96
Příkaz while	97
Příkaz do.....	101
Příkaz break.....	102
Příkaz continue.....	103
Souhrn	103
Cvičení.....	103

Lekce 5

Rozhodování 105

Příkaz if	106
Konstrukce if-else	110
Složené relační testy.....	112
Vnořené příkazy if.....	115
Konstrukce else if.....	116
Příkaz switch	124
Logické proměnné	127
Podmínkový (ternární) operátor	131
Cvičení.....	133

Lekce 6

Třídy zblízka 135

Oddělení rozhraní a implementace do samostatných souborů.....	136
Generované přístupové metody	141
Přístup k atributům pomocí operátoru tečka	142
Metody s více argumenty	143
Metody bez názvů argumentů	145
Operace se zlomky	145
Lokální proměnné	147
Argumenty metod.....	148
Klíčové slovo static.....	149
Klíčové slovo self.....	150
Alokování a vracení objektů z metod.....	151
Rozšíření definic třídy v hlavičkovém souboru.....	155
Cvičení.....	156

Lekce 7

Dědičnost 157

Vše začíná v kořenu.....	158
Nalezení správné metody.....	162
Rozšíření prostřednictvím dědičnosti: Přidání nových metod.....	162
Třída pro bod a alokování paměti	165
Direktiva @class.....	166
Instance vlastníci své objekty	170
Překrývání metod	173
Která metoda se vybere?	175
Překrytí metody dealloc klíčovým slovem super.....	177

Rozšiřování dědičnosti: Doplnění instančních proměnných	179
Abstraktní a sdružené třídy	181
Cvičení.....	182

Lekce 8

Polymorfismus, dynamické typování a dynamická vazba..... 185

Polymorfismus: Stejný název, jiná třída.....	186
Dynamická vazba a typ id	189
Kontrola během kompilace a za běhu	191
Datový typ id a statické typování	192
Typy argumentů a návratových hodnot u dynamického typování	192
Dotazy na třídy	193
Zpracování výjimek pomocí @try	197
Cvičení.....	199

Lekce 9

Více o proměnných a datových typech.....201

Inicializování tříd	202
Návrat k oboru platnosti	204
Direktivy řízení oboru platnosti instančních proměnných	204
Externí proměnné.....	205
Statické proměnné.....	207
Specifikátory paměťové třídy.....	209
auto.....	209
const.....	210
volatile	210
Výčtové datové typy	211
Příkaz typedef	214
Převody datových typů	215
Převodní pravidla.....	215
Rozšíření o znaménko.....	216
Cvičení.....	217

Lekce 10

Kategorie a protokoly219

Kategorie.....	220
Poznámky ohledně kategorií	224
Protokoly	225
Neformální protokoly.....	228

Složené objekty	229
Cvičení.....	230

Lekce 11

Preprocesor.....233

Příkaz #define.....	234
Pokročilejší typy definic	236
Operátor #	239
Operátor ##.....	240
Příkaz #import.....	241
Podmíněná kompilace	243
Příkazy #ifdef, #endif, #else a #ifndef	243
Příkazy preprocesoru #if a #elif	245
Příkaz #undef.....	246
Cvičení.....	246

Lekce 12

Další prvky jazyka C247

Pole	248
Inicializování prvků pole.....	250
Funkce.....	254
Argumenty a místní proměnné.....	255
Vracení výsledku funkce	257
Funkce, metody a pole	260
Struktury	262
Inicializování struktur.....	267
Pole struktur	268
Struktury ve strukturách	268
Další podrobnosti o strukturách	269
Nezapomeňte na objektově orientované programování!.....	272
Ukazatele	272
Ukazatele a struktury	276
Ukazatele, metody a funkce	278
Ukazatele a pole.....	279
Operace s ukazateli	288
Ukazatele a adresy v paměti.....	289
Union.....	290
Nejsou to objekty!	292
Různé prvky jazyka	293
Složené literály.....	293

Příkaz goto	293
Prázdný příkaz.....	294
Operátor čárka	294
Operátor sizeof	294
Argumenty příkazového řádku	295
Jak věci fungují.....	297
Fakt 1: Instanční proměnné jsou uloženy ve strukturách.....	298
Fakt 2: Objektová proměnná je vlastně ukazatelem	298
Fakt 3: Metody jsou funkce a volání metod jsou voláním funkcí.....	299
Fakt 4: Typ id je obecným ukazatelovým typem.....	299
Cvičení.....	299

Část II

Základy frameworku Foundation

Lekce 13

Úvod do frameworku Foundation..... 303

Dokumentace frameworku Foundation.....	304
--	-----

Lekce 14

Čísla, řetězce a kontejnery..... 307

Číselné objekty	308
Rychlý pohled na automaticky uvolňovaný zásobník	310
Textové objekty	312
Více o funkci NSLog	313
Měnitelné a neměnné objekty	314
Měnitelné řetězce.....	319
Kam všechny ty objekty mizí?.....	323
Objekty pole	325
Pozor na hodnotu nil.....	329
Vytvoření adresáře	330
Generované metody třídy Vizitka	334
Rychlé procházení objektových kontejnerů	339
Řazení polí.....	346
Slovníkové objekty	349
Procházení slovníkem	351
Objekty množin.....	353
Cvičení.....	357

Lekce 15

Práce se soubory.....359

Správa souborů a složek: NSFileManager	360
Práce s třídou NSData	365
Práce se složkami.....	366
Procházení obsahem složky	368
Práce s cestami: NSPathUtilities.h	370
Běžné metody pro práci s cestami	372
Kopírování souborů a použití třídy NSProcessInfo.....	374
Základní operace se soubory: NSFileHandle	377
Cvičení.....	382

Lekce 16

Správa paměti.....383

Automaticky uvolňovaný zásobník.....	384
Počítání odkazů	385
Počítání odkazů a řetězce	388
Počítání odkazů a instanční proměnné.....	390
Příklad automatického uvolnění.....	396
Souhrn pravidel správy paměti	398
Automatická správa paměti (garbage collection)	398
Cvičení.....	400

Lekce 17

Kopírování objektů401

Metody copy a mutableCopy	402
Mělké a hluboké kopírování	404
Implementování protokolu <NSCopying>	407
Kopírování objektů v přístupových metodách	409
Cvičení.....	412

Lekce 18

Archivování413

Archivování pomocí seznamů vlastností	414
Archivování pomocí NSKeyedArchiver	416
Zápis kódovacích a dekodovacích metod	418
Použití třídy NSData k vytváření vlastních archivů.....	424
Použití archivace ke kopírování objektů	428
Cvičení.....	429

Část III

Cocoa a SDK iPhone

Lekce 19

Úvod do Cocoa	433
Vrstvy operačního systému.....	434
Cocoa Touch	435

Lekce 20

Psaní aplikací pro iPhone	437
Sada iPhone SDK.....	438
Vaše první aplikace pro iPhone	438
Vytvoření nového projektu aplikace pro iPhone	439
Zadání kódu.....	442
Návrh uživatelského rozhraní	444
Kalkulačka iPhone se zlomky	451
Začátek nového projektu KalkulackaZlomku.....	452
Definování objektu řídicího zobrazovaný rámec.....	454
Třída Zlomek.....	459
Třída Kalkulacka pracující se zlomky	462
Návrh uživatelského rozhraní	463
Souhrn	464
Cvičení.....	465

Část IV

Přílohy

A: Glosář	469
B: Souhrn jazyka Objective-C 2.0.....	475
C: Zdrojový kód adresáře.....	529
D: Prostředky	535
Rejstřík	539

Úvod

Dennis Ritchie v organizaci AT&T Bell Laboratories byl na začátku 70. let 20. století průkopníkem programovacího jazyka C. Tento jazyk se však dočkal širší popularity a podpory až na konci 70. let. To bylo tím, že až do té doby nebyly kompilátory C běžně dostupné pro komerční použití mimo Bell Laboratories. Zpočátku byl nárůst oblíbenosti částečně způsoben také stejným, ne-li rychlejším růstem oblíbenosti operačního systému UNIX, který byl napsán prakticky celý v jazyce C.

Brad J. Cox navrhl jazyk Objective-C na počátku 80. let 20. století. Tento jazyk vycházel z jazyka označovaného za SmallTalk-80. Jazyk Objective-C byl další *vrstvou* jazyka C, což znamená, že k jazyku C byly doplněny různé prvky a vznikl tak nový programovací jazyk umožňující tvorbu *objektů* a manipulaci s nimi.

Společnost NeXT Software si licencovala jazyk Objective-C v roce 1988 a vyvinula jeho knihovny i vývojové prostředí označované NEXTSTEP. V roce 1992 byla podpora Objective-C doplněna do vývojového prostředí Free Software Foundation's GNU. Vlastnická práva ke všem produktům Free Software Foundation (FSF) patří organizaci FSF. Produkt je k dispozici pod GNU General Public License.

V roce 1994 uvolnily společnosti NeXT Computer a Sun Microsystems standardizovanou specifikaci systému NEXTSTEP označovanou jako OPENSTEP. Implementace OPENSTEP od Free Software Foundation se nazývá GNUStep. Linuxová verze, která zahrnuje také linuxové jádro a vývojové prostředí GNUStep, se označuje poměrně trefně jako LinuxSTEP.

20. 12. 1996 oznámila společnost Apple Computer, že přebírá NeXT Software a prostředí NEXTSTEP/OPENSTEP se stalo základem nové zásadní verze operačního systému Apple, OS X. Verze tohoto vývojového prostředí od společnosti Apple bylo pojmenováno Cocoa. S vestavěnou podporou jazyka Objective-C doplněnou vývojovými nástroji jako Project Builder (nebo jeho následníkem Xcode) a Interface Builder vytvořila společnost Apple výkonné prostředí vývoje aplikací na Mac OS X.

V roce 2007 uvolnila společnost Apple aktualizaci jazyka Objective-C, kterou označila za Objective-C 2.0. Zmíněnou verzi jazyka popisuje druhé vydání této knihy.

Když byl v roce 2007 uvolněn iPhone, vývojáři se dožadovali možnosti vyvíjet aplikace pro toto revoluční zařízení. Zpočátku společnost Apple nevívala možnost vývoje aplikací nezávislých tvůrců. Firma alespoň trochu uklidnila potenciální vývojáře pro iPhone tím, že jim umožnila vyvíjet webové aplikace. Webové aplikace běží pod prohlížečem Safari vestavěným v iPhone a vyžadují po uživateli připojení k internetovým stránkám, které provozovanou aplikaci hostí. Vývojáři ovšem nebyli spokojeni s mnoha omezeními danými webovými aplikacemi a společnost Apple krátce nato oznámila, že vývojáři budou moci pracovat také na takzvané *nativních* aplikacích pro iPhone.

Nativní aplikace je taková, která se nachází na zařízení iPhone a běží pod operačním systémem iPhone podobně, jako na něm pracují jiné vestavěné aplikace (jako Contacts, iPod a Weather). OS iPhone

je vlastně verzí Mac OS X, takže aplikace lze vyvíjet a ladit kupříkladu na MacBook Pro. Společnost Apple dokonce brzy nabídla silnou vývojářskou sadu Software Development Kit (SDK), jež umožnila rychlý vývoj a ladění aplikací pro iPhone. Dostupnost simulátoru iPhone dovolila vývojářům ladit aplikace přímo na vývojových systémech, takže programy nebylo nutné stahovat a testovat na skutečném zařízení iPhone nebo iPod Touch.

Co se v knize dozvíte

Když jsem se zamýšlel nad vytvořením učebního materiálu pro Objective-C, musel jsem učinit jedno zásadní rozhodnutí. Svůj text jsem mohl pojmout podobně, jako to činí jiné texty o Objective-C, tedy s předpokladem, že čtenář již umí psát programy v jazyce C. Jazyk jsem mohl také učít z hlediska používání bohaté knihovny rutin, jako jsou frameworky Foundation a Application Kit. Některé knihy také spíše ukazují, jak využívat vývojové nástroje na Macu, jako jsou Xcode a Interface Builder.

Ovšem s přijetím takového přístupu jsem měl problémy. Zprvce, učít se celý jazyk C a teprve potom Objective-C není správné. C je *procedurální* jazyk obsahující mnoho funkcí zbytečných pro programování v Objective-C, zejména na začátečnické úrovni. Ve skutečnosti je využívání některých takových prvků v nesouladu s dobrou metodikou objektově orientovaného programování. Není vhodné seznamovat se s mnoha detaily procedurálního jazyka a teprve poté se věnovat objektově orientovanému jazyku. Programátor začíná nesprávným směrem a získává špatné návyky, ze kterých jen těžko vyplyne dobrý objektově orientovaný styl programování. Samotná skutečnost, že je Objective-C rozšířením jazyka C, neznamená, že se musíte nejprve naučit C.

Proto jsem se rozhodl, že nebudu nejprve učít jazyk C ani nebudu předpokládat jeho znalost. Rozhodl jsem se využít nekonvenční přístup a učít Objective-C a základní jazyk C jako jediný integrovaný jazyk, a to z hlediska objektově orientovaného programování. Smyslem této knihy je přesně její název: Naučit vás programovat v Objective-C 2.0. Netvrdí, že vás podrobně seznámí s používáním dostupných vývojových nástrojů pro vytváření a ladění programů ani že vám poskytne detailní instrukce ohledně vývoje interaktivních grafických aplikací využívajících Cocoa. S tím vším se můžete podrobně seznámit jinde, jakmile budete umět psát programy v Objective-C. Zvládnutí takového materiálu bude ve skutečnosti mnohem jednodušší, budete-li mít slušnou znalost programování v Objective-C. Tato kniha nepředpokládá prakticky žádnou předchozí zkušenost s programováním. Pokud s programováním začínáte, neměli byste mít potíže naučit se Objective-C jako svůj první programovací jazyk.

Tato kniha učí Objective-C na příkladech. Když představím nějakou novou funkci jazyka, obvykle nabídnou malý úplný ukázkový program, který danou funkci ukáže blíže. Obrázek má cenu tisíce slov a totéž platí pro správně vybraný programový příklad. Zásadně vám doporučuji každý program si spustit (všechny jsou k dispozici online) a porovnat výsledky získané na vašem systému s výsledky uvedenými v textu. Tím se seznámíte s jazykem a jeho syntaxí, ovšem zároveň se vám přiblíží proces kompilování a spuštění programů Objective-C.

Jak je kniha uspořádána

Knihy je rozdělena na tři logické části. Část I, „Jazyk Objective-C 2.0“, seznamuje se základy jazyka. Část II, „Základy frameworku Foundation“, ukazuje, jak využít tuto bohatou knihovnu předdefinovaných tříd. Část III, „Programování v Cocoa a iPhone SDK“, nabízí přehled frameworku Application

Kit technologie Cocoa a pak vás provádí procesem vývoje jednoduché aplikace pro iPhone s využitím frameworku UIKit i vývojem a laděním kódu pomocí Xcode a Interface Builder.

Framework (viz poznámka) je sada tříd, rutin a pomocných souborů (zdrojů), jež jsou logicky seskupené tak, aby byl vývoj programů snazší. Značná část možností programování v Objective-C využívá právě rozsáhlých možností nabízených systémovými frameworky.



Poznámka

Anglický termín *framework* je do češtiny často překládán jako aplikační rámec nebo pouze rámec. Tento překlad není právě šťastný, v případě frameworků Cocoa a Cocoa Touch navíc koliduje s rámci jakožto objekty uživatelského rozhraní. Rozhodli jsme se proto termín nepřekládat a zvolili raději jeho počeštěnou variantu, která je v odborné terminologii zcela běžně používána.

Lekce 1, „Programování v Objective-C“, začíná tím, že vám ukáže vytvoření prvního programu v Objective-C.

Protože toto není kniha o programování Cocoa, nejsou tu výrazně rozebírána grafická uživatelská rozhraní (Graphical User Interface – GUI); vlastně se o nich zmiňuji až v části III. Ovšem je zapotřebí nějakým způsobem zadávat vstup a přejímat výstup. Většina příkladů v tomto textu přebírá vstup z klávesnice a vytváří výstup v okně – terminálu, používáte-li `gcc` z příkazového řádku, nebo konzole, používáte-li Xcode.

Lekce 2, „Třídy, objekty a metody“, se zabývá základy objektově orientovaného programování. Tato lekce zavádí určitou terminologii, ovšem jen v minimální míře. Zavádím rovněž mechanismus definování třídy a prostředky pro odesílání zpráv instancím neboli objektům. Instruktoři a zkušení programátoři Objective-C uvidí, že při deklarování objektů využívám *statické* typování. Myslím, že se jedná o nejlepší přístup pro začínající studenty, protože kompilátor pak dokáže zachytit více chyb, programy se lépe dokumentují samy a noví programátoři spíše explicitně deklarují datové typy, jakmile jsou známé. Proto se princip typu `id` a jeho možnosti nepopisují až do lekce 8, „Polymorfismus, dynamické typování a dynamická vazba“.

Lekce 3, „Datové typy a výrazy“, popisuje základní datové typy Objective-C a jejich použití v programech.

Lekce 4, „Programové smyčky“, představuje tři příkazy smyček neboli cyklů, jež můžete v programech využívat: `for`, `while` a `do`.

Rozhodování je zásadním prvkem všech programovacích jazyků. Lekce 5, „Rozhodování“, podrobně popisuje příkazy `if` a `switch` Objective-C.

Lekce 6, „Třídy zblízka“, se noří hlouběji do práce se třídami a objekty. Probírám zde podrobnosti o metodách, více argumentech a místních proměnných.

Lekce 7, „Dědičnost“, představuje základní principy dědičnosti. Tato funkce zjednodušuje vývoj programů, protože můžete využít výhody výše uvedeného. Dědičnost a podřízené třídy usnadňují úpravy a rozšiřování existujících definic tříd.

Lekce 8 probírá základní charakteristiky jazyka Objective-C. Probereme tu polymorfismus, dynamické typování a dynamické vazby.

Lekce 9 až 12 uzavírají popis jazyka Objective-C a zabývají se takovými věcmi, jako je inicializace objektů, protokoly, kategorie, preprocesor i některé skrytější prvky jazyka C včetně funkcí, polí, struktur a ukazatelů. Tyto základní prvky nejsou vždy při vývoji objektově orientovaných aplikací zapotřebí (a často je lepší vyhýbat se jim). Doporučuji vám projít si při prvním čtení lekci 12 „Další prvky jazyka C“, pouze zběžně a vrátit se k ní, jen pokud se budete potřebovat dozvědět více o určité stránce tohoto jazyka.

Část II začíná lekcí 13, „Úvod do frameworku Foundation“, jež přibližuje framework Foundation a přístup k jeho dokumentaci.

Lekce 14 až 18 popisují důležité prvky frameworku Foundation. Sem patří objekty reprezentující čísla a textové řetězce, kontejnery, souborový systém, správa paměti a proces kopírování a archivování objektů.

Jakmile dokončíte studium části II, budete schopni vyvíjet v Objective-C poměrně pokročilé programy využívající framework Foundation.

Část III začíná lekcí 19, „Úvod do Cocoa“. Zde najdete rychlý přehled frameworku Application Kit, jež nabízí třídy nezbytné pro vývoj sofistikovaných grafických aplikací na Macu.

Lekce 20, „Psaní aplikací pro iPhone“, představuje vývojářskou sadu (SDK) pro iPhone framework UIKit. Tato lekce ilustruje v jednotlivých krocích vytvoření jednoduché aplikace pro iPhone (nebo iPod touch). Následuje aplikace kalkulátoru, jež vám umožní používat iPhone k provádění jednoduchých aritmetických výpočtů se zlomky.

Jelikož objektově orientované programování zahrnuje značný objem terminologie, příloha A, „Glosář“, uvádí definice některých běžných termínů.

Příloha B, „Souhrn jazyka Objective-C“, nabízí referenční přehled jazyka Objective-C.

Příloha C, „Zdrojový kód adresáře“, představuje výpis zdrojového kódu dvou tříd vyvíjených a rozsáhle používaných v části II této knihy. Popisovanými třídami jsou vizitka a adresář. Jejich metody dovolují vykonávat jednoduché operace, jako je přidání vizitek do adresáře nebo jejich odstranění, vyhledání nějakého jména, vypsání obsahu adresáře atd.

Jakmile se naučíte psát programy v Objective-C, můžete se pustit několika směry. Máte možnost seznámit se blíže se základy nízkourovňového programovacího jazyka C, ovšem můžete začít také s psaním programů Cocoa běžících na systému Mac OS X či vyvíjet sofistikovanější aplikace pro iPhone. V každém případě vás příloha D, „Zdroje informací“, navede na správnou cestu.

Poděkování

Rád bych poděkoval několika lidem za jejich pomoc při přípravě prvního vydání tohoto textu. Nejprve chci poděkovat Tonymu Ianninovi a Stevenu Levymu za pročetí rukopisu. Jsem vděčný i Miku Gainesovi za jeho připomínky.

Chci také poděkovat technickým redaktorům, Jacku Purdumovi (první vydání) a Miku Trentovi. Měl jsem to štěstí, že Mike Trent redigoval obě vydání této knihy. Byl tím nepečlivějším redaktorem, jaký se kdy stal o mou knihu. Nejenže poukázal na slabiny, ale dokonce přišel s vlastními návrhy. Díky jeho poznámkám k prvnímu vydání jsem změnil přístup k popisu správy paměti a pokusil jsem se

zajistit, aby žádný z ukázkových programů v knize netrpěl „úniky“. Poskytl mi také nedocenitelné připomínky k lekci o programování iPhone.

Již od prvního vydání upravuje Catherine Babinová fotografie na obálce a nabízí mi mnoho hezkých obrázků k výběru. Protože má obálku od přítelkyně, je pro mě tato kniha ještě zvláštnější.

Vděčím také Marku Taberovi z nakladatelství Pearson za to, že mi odpouštěl zpoždění a byl ochoten přepracovávat časový plán a toleroval neustálé posunování konečných termínů odevzdání při práci na tomto druhém vydání. Z nakladatelství Pearson ještě musím poděkovat redaktorovi pro vývoj Michaelu Thurstonu, své textové redaktorce Kristě Hansingové a redaktorce projektu Mandie Frankové, která dokonale zvládla sprint směrem k cíli.

Jsem také velmi vděčný Michaelu de Haanovi a Wendy Muiové za neuvěřitelně pečlivé provedení korektury tohoto druhého vydání. Jejich pozornost k detailům vedla k vytvoření dlouhého seznamu typografických i faktických chyb, jež byly v tomto druhém vydání odstraněny. Vydavatelé, pozor: Tyto dva páry očí mají vysokou cenu!

Jako vždy osvědčily mé děti neuvěřitelnou vyspělost a trpělivost, když jsem tuto knihu v létě sestavoval (a pak pokračoval až do podzimu)! Gregory, Lindo a Julie, miluji vás!

Stephen G. Kochan
říjen 2008