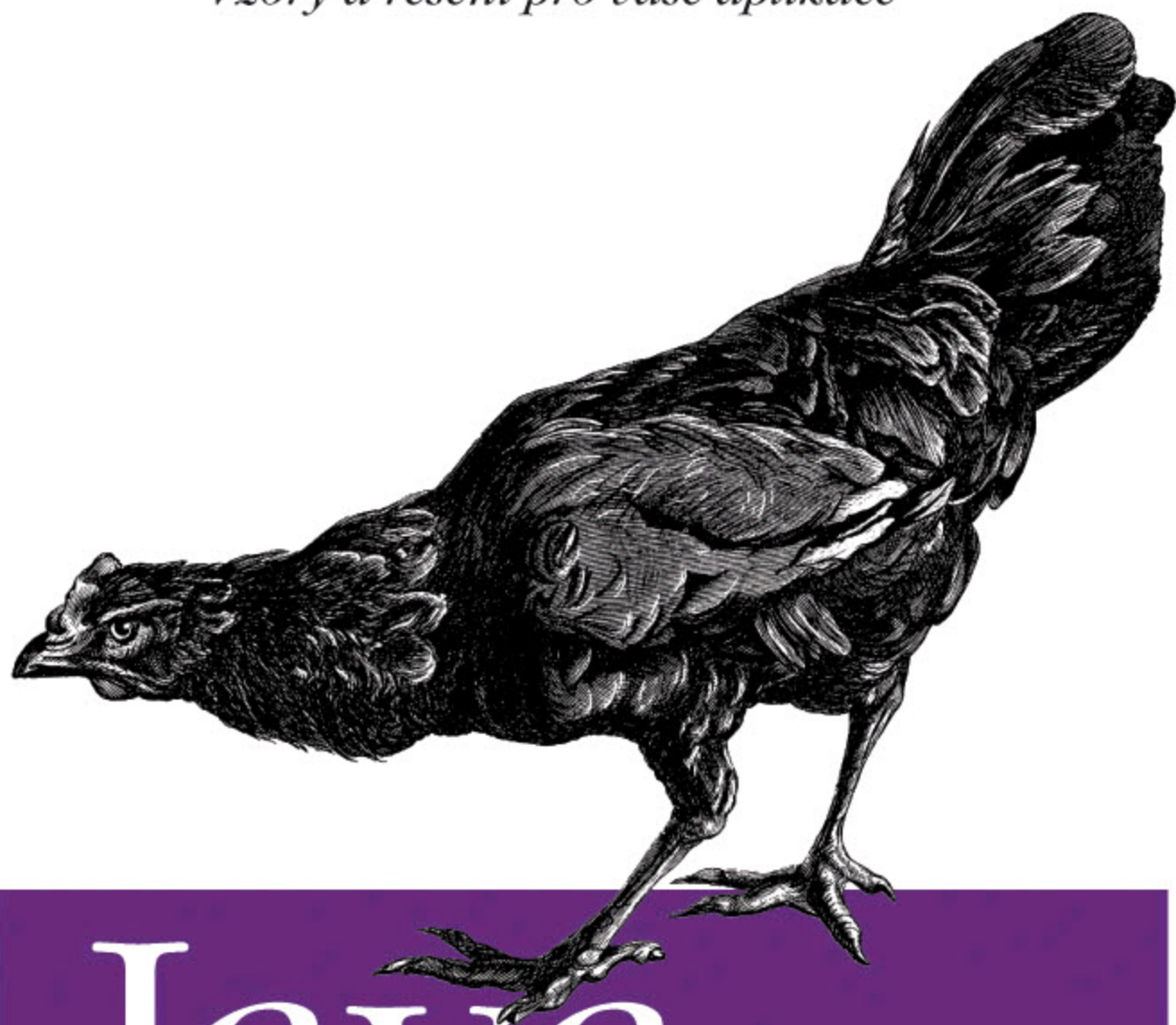


Vzory a řešení pro vaše aplikace



Java™

kuchařka programátora

Ian F. Darwin

O'REILLY®

computer
press

Ian F. Darwin

Java

Kuchařka programátora

Computer Press, a.s.
Brno
2006

Java Kuchařka programátora

Ian F. Darwin

Copyright © Computer Press, a.s. 2006. Vydání první. Všechna práva vyhrazena.
Vydalo nakladatelství Computer Press, a.s. jako svou 2151. publikaci.

Vydavatelství a nakladatelství Computer Press, a.s.,
nám. 28. dubna 48, 635 00 Brno, knihy.cpress.cz

ISBN 80-251-0944-5

Prodejní kód: K1223

Překlad: Jan Gregor

Odborná korektura: Rudolf Pecinovský

Jazyková korektura: Marie Schreinerová

Vnitřní úprava: O'Reilly, Jiří Matoušek

Sazba: Jiří Matoušek, Petr Klíma

Rejstřík: Daniel Štreit

Obálka: O'Reilly

Komentář na zadní straně obálky:

Martin Domes, Ivo Magera

Technická spolupráce: Petr Klíma

Odpovědný redaktor: Martin Domes

Technický redaktor: Jiří Matoušek

Produkce: Petr Baláš

Copyright © Computer Press 2006. Authorized translation of the English edition © 2004 O'Reilly Media, Inc. This translation is published and sold by permission of O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

Authorized translation from English language edition Java Cookbook 2nd edition.

Žádná část této publikace nesmí být publikována a šířena žádným způsobem a v žádné podobě bez výslovného svolení vydavatele.

Computer Press, a.s., nám. 28. dubna 48, 635 00 Brno

tel.: 546 122 111, fax: 546 122 112

Objednávejte na: **knihy.cpress.cz**
distribuce@cpress.cz

Bezplatná telefonní linka: **800 555 513**

Dotazy k vydavatelské činnosti směřujte na: **knihy@cpress.cz**

Máte-li zájem o pravidelné zasílání informací o knižních novinkách do Vaší e-mailové schránky, zašlete nám zprávu, obsahující váš souhlas se zasíláním knižních novinek, na adresu **novinky@cpress.cz**.



Novinky k dispozici ve dni vydání, slevy, recenze,
zajímavé programy pro firmy i koncové zákazníky.

Předmluva **13**

Předmluva k druhému vydání	13
Předmluva k prvnímu vydání	14
Pro koho je tato kniha určena	15
Co v této knize najdete?	16
Poznámky k platformě	18
Jiné knihy	19
Konvence používané v knize	21
Komentáře a otázky	22
Stažení zdrojových kódů	23
Poděkování	24

1. Začínáme: překlad, spuštění a ladění **27**

1.1	Překlad a spuštění programů v Javě: JDK	27
1.2	Upravování a překlad pomocí editoru s podporou barevného zvýrazňování syntaxe	29
1.3	Překlad, spouštění a testování pomocí IDE	30
1.4	Efektivní používání CLASSPATH	36
1.5	Použití tříd API cz.darwinsys z této knihy	38
1.6	Překlad zdrojového kódu příkladů z této knihy	39
1.7	Automatický překlad pomocí nástroje Ant	40
1.8	Spouštění appletů	43
1.9	Zpracování varování o zavržení	44
1.10	Podmíněné ladění bez #ifdef	46
1.11	Ladění výstupů	47
1.12	Udržování korektnosti programu pomocí asercí	49
1.13	Ladění pomocí JDB	50
1.14	Jednotkové testy: vyhněte se potřebě ladících nástrojů	53
1.15	Získejte čitelné zpětné sledování	55
1.16	Hledání dalších zdrojových kódů v Javě	56
1.17	Program: Debug	58

2. Komunikace s prostředím **59**

2.1	Získání proměnných prostředí	59
2.2	Systémové vlastnosti	61
2.3	Psaní kódu závislého na verzi JDK	63
2.4	Psaní kódu závislého na operačním systému	64
2.5	Použití rozšíření nebo dalších zapouzdřených API	66
2.6	Syntaktická analýza argumentů příkazového řádku	67

3. Řetězce **75**

3.1	Rozkládání řetězců do podřetězců	77
3.2	Rozkládání řetězců pomocí StringTokenizer	78
3.3	Spojování řetězců pomocí operátoru +, StringBuilder (JDK 1.5) a StringBuffer	81
3.4	Zpracování řetězce po jednom znaku	83
3.5	Zarovnání řetězců	84
3.6	Převody mezi znaky Unicode a řetězci	86
3.7	Převrácení řetězce po slovech nebo znacích	88
3.8	Rozšiřování a komprese tabulátorů	89
3.9	Ovládání velikosti písmen	94
3.10	Odsazení textových dokumentů	95
3.11	Vkládání netisknutelných znaků	96
3.12	Odstraňování mezer z konce řetězce	97
3.13	Syntaktická analýza dat oddělených čárkou	99
3.14	Program: Jednoduchý formátovač textu	104
3.15	Program: Fonetický porovnávač názvů (soundex)	106

4. Porovnávání vzorů pomocí regulárních výrazů **109**

4.1	Syntaxe regulárních výrazů	111
4.2	Použití regulární výrazů v Javě: Test na vzor	117
4.3	Hledání odpovídajícího textu	121
4.4	Nahrazení odpovídajícího textu	123
4.5	Tisk všech výskytů vzoru	124
4.6	Vypisování řádků obsahující vzory	126
4.7	Ovládání velikosti písmen v regulárních výrazech	127
4.8	Porovnávání „akcentovaných“ nebo složených znaků	129
4.9	Porovnávání nových řádků v textu	130
4.10	Program: Syntaktická analýza protokolového souboru serveru Apache	132
4.11	Program: Dolování dat	133
4.12	Program: Úplný Grep	135

5.1	Kontrola, jestli je řetězec platné číslo	143
5.2	Ukládání větších čísel do menších čísel	144
5.3	Převádění čísel na objekty a naopak	145
5.4	Vytvoření zlomku z celého čísla bez použití čísla s plovoucí desetinnou čárkou	146
5.5	Zajištění přenosnosti čísel s plovoucí desetinnou čárkou	147
5.6	Porovnávání čísel s plovoucí desetinnou čárkou	149
5.7	Zaokrouhlování čísel s plovoucí desetinnou čárkou	151
5.8	Formátování čísel	151
5.9	Převod mezi dvojkovými, osmičkovými, desítkovými a šestnáctkovými čísly	154
5.10	Zpracování skupiny celých čísel	155
5.11	Práce s římskými číslicemi	156
5.12	Správné formátování množného čísla	159
5.13	Generování náhodných čísel	161
5.14	Generování lepších náhodných čísel	162
5.15	Počítání trigonometrických funkcí	164
5.16	Logaritmování	164
5.17	Násobení matic	165
5.18	Používání komplexních čísel	166
5.19	Zpracování velmi velkých čísel	169
5.20	Program: Převodník teplot	171
5.21	Program: Číselné palindromy	174

6.1	Nalezení dnešního datumu	178
6.2	Vypsání datumu/času v daném formátu	179
6.3	Znázornění dat v jiných epochách	182
6.4	Převádění YMDHMS na kalendář nebo sekundy letopočtu	182
6.5	Převod řetězců na data	184
6.6	Převádění sekund epochy na DMYHMS	185
6.7	Sčítání nebo odčítání množství od datumu nebo kalendáře	186
6.8	Rozdíly mezi dvěma daty	187
6.9	Porovnávání dat	188
6.10	Den týdne/měsíce/roku nebo číslo týdne	189
6.11	Vytváření stránky kalendáře	190
6.12	Měření uplynulého času	192
6.13	Dopřejte si chvíli odpočinku	194
6.14	Program: Upomínací služba	195

7. Strukturování dat pomocí Javy **199**

7.1	Použití polí pro strukturování dat	200
7.2	Změna velikosti pole	201
7.3	Dynamičtější pole	202
7.4	Použití iterátorů pro datově nezávislý přístup	204
7.5	Strukturování dat v spojovém seznamu prvků	205
7.6	Mapování pomocí Hashtable a HashMap	207
7.7	Ukládání řetězců do vlastností a preferencí	208
7.8	Seřazení kolekce	212
7.9	Vyhýbejte se nutnosti uspořádávat data	216
7.10	Vyvarujte se duplicitních hodnot	217
7.11	Hledání objektu v kolekci	218
7.12	Převádění kolekce na pole	220
7.13	Definice vlastního iterátoru	220
7.14	Zásobník	222
7.15	Vícerozměrné struktury	223
7.16	Konečně kolekce	225
7.17	Program: Porovnání časové náročnosti	227

8. Novinky ve verzi 5.0 **229**

8.1	Použití parametrizovaných kolekcí	230
8.2	Použití rozšíření cyklů for	231
8.3	Vyhnutí se přetypování pomocí parametrizace typů	232
8.4	Dovolte Javě automatické převody mezi primitivními a jejich obalovými typy	234
8.5	Výčtové typy	235
8.6	Program: MediaInvoicer	239

9. Objektově orientované techniky **243**

9.1	Tisk objektů: Formátování pomocí toString()	245
9.2	Překrytí metody equals	246
9.3	Překrytí metody hashCode	249
9.4	Metoda clone	250
9.5	Metoda finalize	252
9.6	Použití interních, tj. vnořených, vnitřních a lokálních tříd*	254
9.7	Poskytování zpětných volání přes rozhraní	256
9.8	Polymorfismus/abstraktní metody	259
9.9	Předávání hodnot	260
9.10	Aplikace návrhového vzoru Jedináček (Singleton)	263
9.11	Vytváření vlastních výjimek	265
9.12	Program: Plotter	265

10. Vstup a výstup

269

10.1	Načítání ze standardního vstupu	270
10.2	Zapisování do standardního výstupu	274
10.3	Vypisování pomocí třídy Formatter z JDK 1.5	275
10.4	Prohledávání souboru pomocí StreamTokenizer	279
10.5	Prohledávání vstupu pomocí třídy JDK 1.5 Scanner	283
10.6	Otevírání souboru podle názvu	287
10.7	Kopírování souboru	288
10.8	Načítání souboru do řetězce	291
10.9	Přesměřování standardních proudů	291
10.10	Duplikování proudu tak, jak se zapíše	292
10.11	Čtení/zapisování různých znakových sad	294
10.12	Otravné znaky konce řádku	295
10.13	Dejte si pozor na kód závislý na platformě	295
10.14	Čtení „pokračujících“ řádků	296
10.15	Binární data	301
10.16	Hledání	302
10.17	Zapisování datových proudů z C	303
10.18	Ukládání a obnovování objektů Javy	305
10.19	Předcházení výjimkám ClassCastException pomocí serialVersionUID	308
10.20	Čtení a zapisování archivů JAR nebo Zip	309
10.21	Čtení a zapisování komprimovaných souborů	312
10.22	Program: Převod textu na PostScript	313

11. Operace nad adresáři a systémem souborů

317

11.1	Získání informací o souborech	317
11.2	Vytvoření souboru	320
11.3	Přejmenování souboru	321
11.4	Mazání souboru	322
11.5	Vytvoření dočasného souboru	323
11.6	Změna atributů souboru	325
11.7	Procházení adresářové struktury	326
11.8	Získání kořenových adresářů	328
11.9	Vytváření nových adresářů	329
11.10	Program: Find	330

12. Programování externích zařízení: sériové a paralelní porty

333

12.1	Výběr portu	335
12.2	Otevření sériového portu	338
12.3	Otevírání paralelního portu	342
12.4	Řešení konfliktů mezi porty	345

12.5	Čtení a zápis: lock- step	348
12.6	Čtení a zápis: řízené událostmi	350
12.7	Čtení a zápis: vlákna	354
12.8	Program: plotter Penman	356

13. Grafika a zvuk **361**

13.1	Zobrazení grafiky pomocí objektu třídy Graphics	361
13.2	Testování grafických komponent	363
13.3	Zobrazení textu	364
13.4	Zobrazení textu zarovnaného na střed	364
13.5	Zobrazení stínu	365
13.6	Zobrazení textu pomocí tříd z balíčku 2DGraphics	368
13.7	Zobrazení textu s integrovaným typem písma	370
13.8	Zobrazení obrázku	373
13.9	Přehrávání zvukových souborů	377
13.10	Přehrávání videoklipů	378
13.11	Tisk v Javě	381
13.12	Program: PlotterAWT	384
13.13	Program: Grapher	387

14. Grafické uživatelské rozhraní **391**

14.1	Zobrazení komponent GUI	392
14.2	Návrh rozvržení okna	394
14.3	Zobrazení panelu s kartami	397
14.4	Správa událostí: tlačítka	398
14.5	Ošetřování událostí pomocí anonymních vnitřních tříd	400
14.6	Ukončení programu „zavřením okna“	402
14.7	Dialogy: když „později“ je příliš pozdě	406
14.8	Zachycení a formátování výjimek grafického uživatelského rozhraní	408
14.9	Zobrazení programového výstupu v okně	410
14.10	Výběr hodnoty pomocí komponenty JSpinner	413
14.11	Výběr souboru pomocí komponenty JFileChooser	414
14.12	Výběr barvy	417
14.13	Formátování komponent JComponents pomocí HTML	419
14.14	Vystředění hlavního okna	420
14.15	Změna vzhledu a chování programu používajícího pro vytvoření GUI třídy knihovny Swing	421
14.16	Rozšíření grafického uživatelského rozhraní pro Mac OS X	425
14.17	Program: upravený dialog pro výběr písma	427
14.18	Program: upravený správce rozvržení	431

15. Podpora cizích jazyků a lokalizace

439

15.1	Tvorba tlačítka pomocí prostředků I18N	439
15.2	Výpis dostupných národních prostředí	441
15.3	Vytváření nabídky pomocí prostředků I18N	442
15.4	Vytváření konvenčních rutin pro internacionalizaci	443
15.5	Vytváření dialogového okna pomocí prostředků I18N	445
15.6	Vytváření svazku prostředků (resource bundle)	446
15.7	Vytahování řetězců ze zdrojového kódu	447
15.8	Nastavení odlišného národního prostředí	448
15.9	Nastavení výchozího národního prostředí	449
15.10	Formátování textu zprávy	450
15.11	Program: MenuIntl	452
15.12	Program: BusCard	454

16. Síťoví klienti

459

16.1	Navázání spojení se serverem	461
16.2	Hledání a oznamování síťových adres	462
16.3	Ošetřování síťových chyb	463
16.4	Čtení a zápis textových dat	464
16.5	Čtení a zápis binárních dat	466
16.6	Čtení a zápis serializovaných dat	468
16.7	Datagramy protokolu UDP	470
16.8	Program: Klient TFTP implementovaný pomocí protokolu UDP	472
16.9	Program: klient programu Telnet	476
16.10	Program: Klient pro chat	479

17. Serverová Java: sokety

485

17.1	Uvedení serveru do provozu	485
17.2	Odeslání odpovědi (textové i binární)	488
17.3	Odeslání informací pomocí objektů	491
17.4	Zpracování více klientů	492
17.5	Obsluha protokolu HTTP	497
17.6	Zabezpečení webového serveru pomocí SSL a JSSE	499
17.7	Protokolování síťového provozu	501
17.8	Protokolování sítě pomocí log4j	506
17.9	Protokolování v síti pomocí JDK 1.4	508
17.10	Hledání síťových rozhraní	510
17.11	Program: Chatový server v Javě	511

18. Síťoví klienti II: Aplety a weboví klienti **517**

18.1	Vložení Javy do webové stránky	517
18.2	Techniky vytváření apletů	519
18.3	Kontaktování serveru na hostitelském počítači apletu	521
18.4	Zobrazení dokumentu pomocí apletu	524
18.5	Spuštění JavaScriptu pomocí apletu	526
18.6	Spuštění skriptu CGI pomocí apletu	527
18.7	Čtení obsahu z dané adresy URL	528
18.8	URI, URL nebo URN?	529
18.9	Vytahování značek HTML z URL	530
18.10	Vytahování adres URL ze souboru	533
18.11	Převod názvu souboru na adresu URL	534
18.12	Program: MkIndex	535
18.13	Program: LinkChecker	540

19. Java a elektronická pošta **547**

19.1	Zasílání e-mailu: Verze pro prohlížeč	547
19.2	Opravdové zasílání e-mailů	552
19.3	Serverový program s podporou pro e-maily	554
19.4	Zasílání e-mailů v kódování MIME	558
19.5	Poskytování nastavení e-mailu	560
19.6	Zasílání e-mailů bez použití JavaMail	561
19.7	Čtení e-mailů	566
19.8	Program: MailReaderBean	570
19.9	Program: MailClient	573

20. Databázový přístup **585**

20.1	Jednoduchý databázový přístup pomocí JDO	586
20.2	Textové databáze	589
20.3	Databáze DBM	593
20.4	Nastavení JDBC a připojení	596
20.5	Připojení k databázi JDBC	598
20.6	Zasílání dotazu JDBC a získávání výsledků	601
20.7	Použití připravených příkazů JDBC	604
20.8	Použití uložených procedur pomocí JDBC	608
20.9	Modifikace dat pomocí ResultSet	608
20.10	Uchovávání výsledků v RowSet	609
20.11	Modifikace dat pomocí SQL	611
20.12	Hledání metadat JDBC	613
20.13	Program: SQLRunner	617

21. XML

629

21.1	Generování XML z objektů	631
21.2	Transformování XML pomocí XSLT	632
21.3	Syntaktická analýza XML pomocí SAX	635
21.4	Syntaktická analýza XML pomocí DOM	637
21.5	Ověřování struktury pomocí DTD	641
21.6	Generování vlastního XML pomocí DOM	642
21.7	Program: xml2mif	644

22. Distribuovaná Java: RMI

647

22.1	Definování podmínek RMI	648
22.2	Vytváření klienta RMI	650
22.3	Vytváření serveru RMI	651
22.4	Zavádění RMI přes síť	653
22.5	Program: Zpětné volání RMI	654
22.6	Program: NetWatch	657

23. Balíčky a balení

665

23.1	Vytváření balíčku	666
23.2	Dokumentace tříd pomocí Javadoc	667
23.3	Za hranice JavaDoc: Anotace/Metadata (JDK 1.5) a XDoclet	671
23.4	Archivace do souborů jar	673
23.5	Spouštění apletu z archívu JAR	674
23.6	Spuštění apletu pomocí moderního JRE	675
23.7	Spouštění aplikace z archívu JAR	678
23.8	Příprava třídy jako JavaBean	680
23.9	Vkládání komponenty JavaBean do archívu JAR	684
23.10	Balení serveletu do souboru WAR	685
23.11	„Jedenkrát vytvořte a instalujte kdekoli“	685
23.12	„Jednou vytvořte a instalujte na systémy Mac OS X“	686
23.13	Java Web Start	688
23.14	Podepisování vašeho souboru JAR	693

24. Práce s vlákny

695

24.1	Spouštění kódu v odlišném vlákně	696
24.2	Zobrazení pohyblivého obrázku pomocí animace	700
24.3	Zastavení vlákna	704
24.4	Rendezvous vláken a časové limity	706
24.5	Synchronizace vláken pomocí klíčového slova synchronized	707
24.6	Zjednodušování synchronizace pomocí zámků JDK 1.5	712

24.7	Synchronizace vláken pomocí wait(?) a notifyAll(?)	716
24.8	Zjednodušení modelu producent-konzument pomocí rozhraní Queue sady JDK 1.5	721
24.9	Automatické ukládání v editoru	724
24.10	Program: Síťový server s vlákny	725
24.11	Zjednodušení serverů pomocí Concurrency Utilities (JDK 1.5)	732

25. Reflexe **735**

25.1	Deskriptor tříd	736
25.2	Hledání a používání metod a proměnných	737
25.3	Dynamické zavádění tříd a vytváření instancí	740
25.4	Vytváření třídy úplně od začátku	742
25.5	Měření výkonu	743
25.6	Vypisování informací o třídě	746
25.7	Program: CrossRef	748
25.8	Program: AppletViewer	753

26. Použití Javy s jinými programovacími jazyky **761**

26.0	Úvod	761
26.1	Spuštění programu	761
26.2	Spuštění programu a zachytávání jeho výstupu	765
26.3	Propojení Javy se skripty pomocí rozhraní BSF	768
26.4	Kombinace jazyků Java a Perl	772
26.5	Použití nativního kódu (jazyky C/C++)	776
26.6	Volání Javy z nativního kódu	781
26.7	Program: DBM	782

27. Doslov **785**

Rejstřík **787**

Předmluva k druhému vydání

Vzrušující změnu na území Javy přináší vývojářská sad a JDK 1.5 s krycím názvem Tiger. Tato verze uvádí několik nových hlavních schopností, například generické typy pro lepší strukturování dat, metadata pro opatřování tříd Java™ komentáři flexibilním, ale přesně stanoveným způsobem, nové mechanismy pro čtení dat založené na vzorech a nový mechanismus pro formátovaný tisk. JDK 1.5, která je nezbytností pro vývojáře Javy, tvoří kromě těchto hlavních změn i mnohem větší množství menších avšak důležitých změn. Někaký čas ještě uplyne, než budou tyto mechanismy plně pochopeny a dostanou se do širšího oběhu, nicméně poté o nich budete chtít okamžitě vědět.

V doslovu prvního vydání jsem uvedl, že „psaní této knihy bylo ponižující zkušeností“. Měl bych tedy doplnit, že také její údržba byla ponižující. I když celá řada recenzentů a autorů nešetřila chválou – jeden velmi laskavý recenzent ji dokonce označil za „prokazatelně nejlepší knihu, která kdy byla o programovacím jazyce Java napsána“ – byl jsem zahanben počtem chyb a opomenutí v prvním vydání. Při přípravě druhého vydání jsem se snažil všechny nedostatky napravit.

Současně jsem doplnil celou řadu nových návodů a menší počet starých návodů jsem odstranil. Největší samostatný doplněk představuje kapitola 8, v níž se zabýváme generickými typy a výčty, funkcionalitami, které poskytují zvýšenou flexibilitu pro kontejnery, jako například Java Collections. Nyní, když Java zahrnuje aplikační programové rozhraní pro regulární výrazy, se kapitola 4 změnila z API pro regulární výrazy Apache na Regulární výrazy JDK 1.4.

Poněkud váhavě jsem odstranil kapitolu Síťový web, která obsahovala program JabaDot Web Portal Site. Jednalo se o nejdelší samostatný příklad programu v knize, který vyžadoval značné předělání (ve skutečnosti bylo třeba tento příklad kompletně přepsat). Kdybychom psali takové webové stránky nyní, mnohem více bychom využili značky JSP a téměř určité bychom pracovali s nějakým webovým frameworkem, jako je například Struts (<http://jakarta.apache.org/struts>), SOFIA (<http://www.salmonllc.com/>) nebo Spring Framework (<http://www.springframework.org/>), abychom se vyhnuli značnému množství zdlouhavého programování. Případně bychom mohli využít nějaký stávající balík – například JLCF od Java Lobby. Informace o Servletech a JavaServer Pages lze najít v knize vydavatelství O'Reilly *Java Servlet & JSP Cookbook* od Bruce W. Perryho. Pojednání o samotném Struts obsahuje kniha Chucka Cavanesse *Programming Jakarta Struts* (O'Reily). Webové služby založené na SOAP objasňuje kniha *Java Web Services* od Dave Chappella a Tylera Jewella (O'Reily), takže tímto tématem se zde nebudeme zabývat.

I když jsme příklady testovali na různých systémech a pro znovusestavení všech kódů poskytujeme skripty Ant, převážnou část nového vývoje a textů k tomuto vydání jsem dělal na systému Mac OS X, který je skutečně „Unixem pro širokou veřejnost“ a který poskytuje jednu z nejlépe podporovaných platform pro práci s Javou. Mac OS X Java však trochu trpí syndromem „zpoždování nových verzí“, protože v době, kdy šlo druhé vydání knihy do tisku, verze 1.5 nebyla pro Macintosh dostupná. Materiál JDK 1.5 byl vyvíjen a testován na Linuxu a Windows.

Rád bych vyjádřil své upřímné poděkování všem, kteří po vytištění prvního anglického vydání * zaslali komentáře nebo kritiku knihy. Zejména musím uvést jednoho z německých překladatelů knihy, Gisberta Selkeho, který během svého překladu přečetl celé první vydání knihy a zjednodušil mou angličtinu. Totéž udělal znovu i pro druhé vydání a rovněž přetvořil celou řadu programových kódů, čímž udělal tuto knihu mnohem lepší, než by jinak byla. Gisbert dokonce zašel za hranice svých povinností a přispěl do knihy jedním návodem (návod 26.4). Dále přepracoval některé jiné návody v téže kapitole. Děkujeme, Gisberte! Druhé vydání také obohatily komentáře Jima Burgesse, který přečetl značnou část knihy. K jednotlivým kapitolám jsme obdrželi komentáře od Jonathana Fuertha, Kima Fowlera, Marca Loya a Mika McCloskeye. Některé kapitoly byly také zkorigovány mou ženou Betty a našimi dospívajícími dětmi.

Odhalením závažných chyb nebo navržením zlepšení oproti prvnímu vydání přispěli do knihy tito lidé: Rex Bosma, Rod Buchanan, John Chamberlain, Keith Goldman, Gilles-Philippe Gregoire, B. S. Hughes, Jeff Johnston, Rob Konigsberg, Tom Murtagh, Jonathan O'Connor, Mark Petrovic, Steve Reisman, Bruce X. Smith a Patrick Wohlwend. Všem patří moje díky a omlouvám se, pokud jsem na někoho zapomněl.

Mé díky rovněž patří lidem v diskusním fóru nakladatelství O'Reilly „Otázky týkající se knih“ za poskytování nenacvičených odpovědí na tolik otázek. Za redakční a produkční práci na druhém vydání děkuji Mikovi Loukidesovi, Deb Cameron a Marlowe Shaeffer.

Předmluva k prvnímu vydání

Znáte-li trochu Javu, je to skvělé. Znáte-li Javu trochu více, je to ještě lepší! Tato kniha je ideální pro čtenáře, kteří mají základní znalosti o jazyce Java a chtějí se něco nového naučit. Nevíte-li zatím o Javě nic, měli byste si nejprve přečíst některou z knih, které se více zaměřují na základy tohoto programovacího jazyka. Nakladatelství O'Reilly například vydalo *Head First Java* or *Learning Java*, pokud se poprvé setkáváte s touto rodinou jazyků nebo *Java in a Nutshell*, pokud patříte ke zkušeným programátorům v jazyce C.

V C jsem začal programovat v roce 1980, když jsem pracoval na Torontské univerzitě, a v průběhu osmdesátých a devadesátých let dvacátého století mi jazyk C docela dobře sloužil. V roce 1995, když se rodící jazyk Oak přejmenoval na Javu, se na mě usmálo štěstí a můj kolega, J. Greg Davidson, mi o novém jazyce pověděl. Poslal jsem e-mail na adresu, kterou mi Greg doporučil, a v březnu roku 1995 jsem obdržel od Jamese Goslinga, tvůrce Javy, následující odpověď:

```
> Zdravím. Přítel mi řekl o Vašem rozšiřitelném síťovém prohlížeči  
> WebRunner(?). WebRunner a Oak(?), jeho jazykové rozšíření, nezní to skvěle? Můžete
```

* První vydání je v současnosti k dispozici v angličtině, němčině, francouzštině, polštině, ruštině, korejštině, tradiční čínštině a zjednodušené čínštině.

> mi prosím sdělit, jestli je již dostupný k vyzkoušení a/nebo jestli
> je k němu na FTP dostupné nějaké odborné pojednání?

Vyzkoušejte <http://java.sun.com>

(Oak se přejmenoval na Java a WebRunner se přejmenoval na
HotJava, aby byli právníci spokojeni)

Stáhl jsem si HotJava a začal jsem ji zkoušet. Zpočátku jsem byl trochu skeptický k tomuto modernímu jazyku, který vypadal jako zkomolenina jazyků C/C++. Napsal jsem testovací a ukázkové programy, několik jsem jich najednou vložil do adresáře, který jsem označil jako *javasrc*, abych uchovával tyto soubory odděleně od mých zdrojových kódů v C (protože programy měly mnohdy stejný název). Jakmile jsem však trochu pronikl do tajů jazyka Java, začal jsem objevovat její výhody pro celou řadu různých činností, například pro automatické obnovení paměti a eliminaci kalkulací ukazatele. Můj adresář *javasrc* se stále zvětšoval, a když jsem napsal kurzy Javy pro společnost Learning Tree*, dostal se do bodu, kdy bylo potřeba vytvořit podadresáře. I poté bylo stále obtížnější něco v adresáři najít, takže brzy bylo zřejmé, že budu potřebovat nějaký typ dokumentace.

Tato kniha je v určitém smyslu výsledkem rychlé kolize mezi mým adresářem *javasrc* a dokumentačním rámcem zřízeným pro další nově vznikající jazyk. Tom Christiansen a Nathan Torkington vytvořili v knize nakladatelství O'Reilly *Perl Cookbook* velmi úspěšnou strukturu, v níž se předkládá materiál v krátkých, účelově zaměřených článcích označovaných jako „návod“ neboli jakési „recepty“. Výchozím modelem této knihy je pochopitelně všem důvěrně známá kuchařka. Použití pojmu „kuchařka“ pro označení výčtu návodů, „jak něco udělat“ na počítači, není žádnou žhavou novinkou. Analogii „kuchařky“ aplikoval na straně softwaru Donald Knuth ve své knize *The Art of Computer Programming* (Addison Wesley), poprvé vydané v roce 1968. Na straně hardwaru napsal Don Lancaster knihu *The TTL Cookbook* (Sams, 1974). (TTL neboli tranzistorové tranzistorová logika představovala v té době malé stavební bloky elektronických obvodů.) Tom a Nathan vypracovali úspěšnou variaci tohoto konceptu a jejich knihu doporučuji každému, kdo se chce dozvědět více o jazyce Perl. No a práce, kterou právě čtete, se snaží být knihou pro lidi, kteří se chtějí dozvědět více o jazyce Java.

Kód v každém návodu by měl být do značné míry samostatný; jakékoli části příkladů si můžete bez obav vypůjčit pro využití ve vlastních projektech. Kód je distribuován s berkeleyovskými autorskými právy, pouze abychom zabránili obchodní reprodukci.

Pro koho je tato kniha určena

Předpokladem k úplnému pochopení výkladu knihy je, že máte základní znalosti o jazyce Java. Nebudeme vám vysvětlovat, jak současně vypsát pomocí `println` řetězec a číslo nebo jak napsat třídu, která rozšiřuje `Applet` a v okně vypíše vaše jméno. Předpokládáme, že jste absolvovali kurz o Javě nebo jste studovali nějakou knihu vysvětlující základy Javy, například *Head First Java*, *Learning Java* nebo *Java in a Nutshell* z vydavatelství O'Reilly. V kapitole 1 si však vysvětlíme určité techniky, které byste nemuseli moc dobře znát a které jsou nezbytné k pochopení později uváděných informací. Neobávejte se přeskakovat jednotlivé části! Jak tiš-
těná, tak elektronická verze knihy obsahuje množství křížových odkazů.

* Jedna z předních technologicky vyspělých nezávislých vzdělávacích společností na světě; viz <http://www.learningtree.com/>.

Co v této knize najdete?

Narozdíl od mých kolegů Toma a Nathana nemusím tolik času věnovat zvláštnostem a idiomům jazyka: Java je příjemně novým způsobem zbavena zvláštních kliček. To však neznamená, že její dobré zvládnutí je jednoduché! V tom případě by totiž nebylo třeba této knihy. Naše hlavní koncepce spočívá tedy v zaměření na aplikační rozhraní Javy. Pomocí příkladů vás naučíme, jaké jsou aplikační rozhraní API a k čemu jsou dobrá.

Java je podobně jako Perl jazyk, na který si zvyknete a který poroste s vámi. A osobně musím připustit, že v současnosti většinou používám Javu. Věci, které bych dříve psal v C, jsou nyní – s výjimkou ovladačů zařízení a zastaralých systémů – udělány v Javě.

Java je však vhodná pro jiný rozsah úloh než jazyk Perl. Perl (a další skriptovací jazyky, například awk a Python) se hodí zejména pro „one-liner“ obslužné úlohy. Jak Tom a Nathan ve své knize předvádí, Perl je vynikající pro úlohy jako např. tisk 42. řádku ze souboru. Tyto věci lze pochopitelně udělat i v Javě, nicméně jelikož Java je kompilovaný, objektově orientovaný jazyk, zdá se být vhodnější pro „vývoj ve velkém“ nebo vývoj podnikových aplikací. Na tento typ vývoje se ve skutečnosti zaměřila řada aplikačních rozhraní API doplněných v Java 2. Spoustu technik si však nevyhnutelně předvedeme na kratších příkladech, a dokonce fragmentech kódu. Buďte si jisti, že každý řádek kódu, který v této knize uvidíte, byl zkompilován a spuštěn.

Celá řada delších příkladů v této knize jsou nástroje, které jsem původně napsal k automatizaci nějaké světské úlohy nebo něčeho jiného. Program `MkIndex` (popsaný v kapitole 17) například načítá nejvyšší adresář místa, kde uchovávám všechny své příklady zdrojových kódů v Javě, a pro tento adresář sestaví z hlediska prohlížeče přívětivý soubor `index.html`. U jiného příkladu bylo tělo prvního vydání zčásti sestaveno v XML, zjednodušení, které se spoléhá na léta zkušeností v SGML (původní standard, který vedl ke značkové syntaxi jazyka HTML). V tomto bodě není zřejmé, jestli bude XML primárně užitečné jako publikační formát nebo jako formát na zpracování dat, případně jestli jeho rozmach bude dále zastírat tento rozdíl, ačkoli se zdá, že zastírání rozdílů je pravděpodobnější. V této knize však formát XML používáme k vepsávání a vyznačování původního textu některých kapitol. Text byl poté převeden do publikačního formátu pomocí programu `XmlForm`. Tento program rovněž ošetřuje – pomocí dalšího programu, `GetMark` – úplné a částečné vložení kódu ze zdrojového adresáře. O `XmlForm` budeme hovořit v návodu 21.7.

Nyní si stručně projdeme uspořádání této knihy. Nejprve si v kapitole 1, *Začínáme: kompilace, spuštění a ladění*, popíšeme určité metody kompilování vašeho programu na různých platformách, spouštění programů v různých prostředích (v prohlížeči, na příkazovém řádku, v okně na pracovní ploše) a ladění. V kapitole 2, *Komunikace s prostředím*, přejdeme od kompilování a spouštění vašeho programu k jeho přizpůsobování okolnímu prostředí – jiným programům, které jsou na vašem počítači.

Následujících pět kapitol pojednává o základním aplikačním rozhraní. V kapitole 3, *Řetězce*, se zaměříme na jeden z nejzákladnějších, ale výkonných datových typů v Javě. Ukážeme si, jak sestavit, rozložit, porovnat a přeuspořádat to, co byste jinak mohli považovat za obyčejný text.

V kapitole 4, *Porovnávání vzorů pomocí regulárních výrazů*, vás naučíme, jak při porovnání řetězců a porovnání vzorů v celé řadě problematických oblastí používat mocné regulární výrazy, technologii pocházející ze systému Unix. Prvním vydáním JDK, které nabízelo tuto moc-

nou technologii, byla verze 1.3; rovněž se zmíníme o několika balících regulárních výrazů třetích stran.

V kapitole 5, *Čísla*, budeme hovořit o vestavěných typech, například `int` a `double`. Také se zmíníme o odpovídajících třídách API (`Integer`, `Double` atd.) a převodových a testovacích schopnostech, jež nabízí. Krátce se rovněž zmíníme o třídách pro „velká čísla“. Jelikož programátoři Javy často potřebují zpracovávat data a časy, a to jak v místním, tak v mezinárodním formátu, probereme si toto důležité téma v kapitole 6, *Data a časy*.

Další dvě kapitoly pojednávají o zpracování dat. V Javě, jako ve většině jazyků, jsou *pole* lineární, indexové kolekce podobných druhů objektů, o čemž budeme hovořit v kapitole 7, *Strukturování dat pomocí Javy*. V této kapitole pokračujeme v diskusi o celé řadě tříd „kolekcí“: výkonných způsobech uchovávání množství objektů v balíku `java.util`. Do této části byla v druhém vydání přidána nová kapitola. JDK 1.5 představila nový rozměr v pojetí strukturování dat a to převzetím konceptu šablon C++ do Java Collections; výsledek označovaný jako generičnost je hlavním námětem kapitoly 8, *Strukturování dat pomocí generičnosti, foreach a výčtů (JDK 1.5)*.

Navzdory určité syntaktické podobnosti s procedurálními jazyky jako C je Java ve své podstatě objektově orientovaný jazyk. V kapitole 9, *Objektově orientované techniky*, si probereme určité klíčové koncepty OOP (objektově orientovaného programování), jak se týkají Javy včetně obvykle předefinovaných metod třídy `java.lang.Object`, a důležité otázky návrhových vzorů.

Několik dalších kapitol pojednává o aspektech tradičního vstupu a výstupu. V kapitole 10, *Vstup a výstup*, si podrobně vysvětlíme pravidla pro načítání a zapisování do souborů. (Nepřeskakujte tuto kapitolu, pokud si myslíte, že soubory jsou nudné, protože některé informace budete potřebovat v nadcházejících kapitolách: v kapitole 12 budete číst a zapisovat na sériových nebo paralelních portech a na soketovém síťovém připojení v kapitole 16!) V kapitole 11, *Ope- race nad adresáři a systémem souborů*, si ukážeme všechno ostatní o souborech – například hledání jejich velikost, a času poslední modifikace – a o čtení a modifikaci adresářů, vytváření dočasných souborů a přejmenovávání souborů na disku. V kapitole 12, *Programování externích zařízení: sériové a paralelní porty*, vám ukážeme, jak můžete používat API `javax.comm` pro čtení-zapisování na sériových a paralelních portech pomocí standardního API Javy.

V kapitole 13, *Grafika a zvuk*, se podíváme na vývoj grafického uživatelského rozhraní (GUI). Tato kapitola představuje směsici informací o činnostech na nižší úrovni, například kreslení grafiky a nastavování typu písma a barev, ale také činnostech na velmi vysoké úrovni, například ovládání videoklipů nebo filmů. V kapitole 14, *Grafická uživatelská rozhraní*, se budeme zabývat aspekty GUI na vyšší úrovni, například tlačítky, návěštími, nabídkami a dalšími – předefinovanými komponentami GUI. Jakmile získáte GUI (ve skutečnosti dříve, než jej opravdu napíšete), budete si chtít přečíst kapitolu 15, *Podpora cizích jazyků a lokalizace*, protože vaše programy mohou fungovat v Akbaru, Afganistanu, Alžíru, Amsterdamu nebo Angleterru stejně tak dobře jako v Albertě, Arkansasu nebo Alabamě...

Jelikož Java byla původně publikována jako „programovací jazyk pro Internet“, je pouze spravedlivé, že nějaký čas věnujeme práci se sítí. V kapitole 16, *Síťové klienti*, budeme pojednávat o základech síťového programování ze strany klienta se zaměřením na sokety. Poté se v kapitole 17, *Serverová Java: Sokety*, přesuneme na stranu serveru. V kapitole 18, *Síťové klienti II: Aplety a weboví klienti*, vás naučíme další techniky prováděné na straně klienta. Programy na Internetu často musí generovat e-maily, takže tato část končí kapitolou 19, *Java a elektronická pošta*.

V kapitole 20, *Databázový přístup*, si vysvětlíme základy balíků JDBC (Java Database Connectivity) a JDO (Java Data Objects). Ukážeme vám, jak se můžete připojit k místním nebo vzdáleným relačním databázím, ukládat a vybírat data a zjišťovat informace o výsledcích dotazu nebo o databázi.

Další tvar ukládání a výměny data nabízí jazyk XML. V kapitole 21, *XML*, si probereme formáty XML a některé operace, které můžete použít pomocí dvou standardních API Javy: SAX a DOM.

V kapitole 22, *Distribuovaná Java: RMI*, pokročíme s konceptem distribuce o krok vpřed a vysvětlíme si RMI (Remote Method Invocation), mechanismus vzdáleného volání procedur Javy. RMI vám dovolí sestavovat klienty, servery, a dokonce scénáře „zpětného volání“ pomocí standardního mechanismu Javy – rozhraní – pro popis dohody mezi klientem a serverem.

V kapitole 23, *Balíky a balíkování*, si ukážeme, jak vytvářet balíky tříd, které vzájemně spolupracují. V této kapitole budeme také hovořit o „implementaci“ nebo distribuci a instalaci vašeho softwaru.

V kapitole 24, *Zpracování vláken v Javě*, se naučíme psát třídy, které vypadají, že dělají více úloh najednou a dovolí vám využít výkonu víceprocesorového hardwaru.

V kapitole 25, *Pobled dovnitř neboli introspekce*, vám odhalíme tajemství, jak mechanicky psát dokumenty API s křížovými odkazy („staňte se ve svém volném čase věhlasnými autory knihy o Javě!“) a jak jsou webové prohlížeče schopny načítat jakýkoli starý aplet – aniž by kdy předtím spatřily tuto specifickou třídu – a spustit ho.

Někdy již máte kód napsaný a pracujete v jiném jazyce, který za vás může udělat část práce nebo chcete použít Javu jakou součást většího balíku. V kapitole 26, *Použití Javy s jinými jazyky*, vám ukážeme, jak spustit externí program (kompilovaný nebo skript) a také přímo komunikovat s „nativním kódem“ v C/C++ v jiných jazycích.

V této 800stránkové knize není místo na všechno, co bych vám rád o Javě sdělil. V *Doslovu* zmiňujeme několik závěrečných myšlenek a odkaz na moje on-line shrnutí aplikačních rozhraní Javy, o nichž by měl vědět každý vývojář Javy.

Žádní dva programátoři nebo autoři se neshodnou na nejvhodnějším pořadí uvedení všech témat Javy. Abychom vám pomohli vždy najít to, co právě hledáte, zahrnuli jsme do knihy četné křížové odkazy, v nichž se odvoláváme na příslušnou část většinou podle čísla návodu.

Poznámky k platformě

Doposud Java prošla pěti hlavními verzemi. První oficiální verzí byla JDK 1.0 a poslední opravnou verzí byla 1.0.2. Druhou hlavní verzí je Java JDK 1.1 a poslední opravnou verzí je 1.1.9, ačkoli v době, kdy budete číst tuto knihu, může být toto číslo již vyšší. V prosinci 1998 vyšla třetí hlavní verze s označením JDK 1.2, ale někdo ze společnosti Sun ji v době jejího vydání neočekávaně přejmenoval na Java 2 a implementace se nazývá jako Java 2 SDK 1.2. Aktuální verzí v době psaní prvního vydání této knihy byla Java 2 SDK 1.3 (JDK 1.3), jež byla publikována v roce 2000.

V době, kdy šlo první vydání této knihy do tisku, se objevila verze Java 2 SDK 1.4; při dokončování knihy vyšla beta verze (kterou společnost Sun označila za „prvotní přístup“), proto jsem ji mohl uvést pouze stručně. Zdá se, že druhé vydání této knihy má lepší načasování; verze Java 2 SDK 1.5 je při provádění aktualizace knihy ve stadiu beta-verze.

V této knize se zaměříme na pátou verzi J2SE (Java 2 Standard Edition), čili verzi 1.5. Předpokládáme, že v době publikace knihy budou všechny vývojové projekty Javy používat JDK 1.4 s velmi malým množstvím spojení na předchozí verze z historických důvodů. K testování kódu jsem kvůli přenositelnosti používal několik platform. Testy jsem prováděl pomocí Linux JDK od Sunu. Pro masový trh jsem otestoval celou řadu programů na implementaci Win32 (Windows 2000/XP/2003) od Sunu. A „pro nás ostatní“ jsem provedl velkou část mého nedávného vývoje pomocí Mac OS X verze 10.2x a novějších verzí od Apple. Jelikož je však Java přenositelná, očekávám, že drtivá většina příkladů bude fungovat na jakékoli platformě podporující Javu, kromě platform vyžadujících zvláštní API. Všechny příklady nebyly testovány na každé platformě, ale všechny byly testovány alespoň na jedné platformě – a většina příkladů na více platformách.

Java API se skládá ze dvou částí: základní API (Core API) a doplňkové API. Základní API je samozřejmě to, které je obsaženo v JDK, jež si zdarma stáhnete z webových stránek <http://java.sun.com/>. Všechno ostatní je doplňkové. Přesto však toto „základní“ API není vůbec malé: čítá kolem 50 balíků, hodně přes 2 000 veřejných tříd a na každou třídu v průměru připadá asi 12 veřejných metod. Programy, které se drží tohoto základního API, mají přiměřeně zajištěnou přenositelnost na jakoukoli platformu Javy.

Doplňkové API se dále dělí na standardní rozšíření a nestandardní rozšíření. Všechny názvy balíčků standardních rozšíření mají prefix `javax.*` (a Sun poskytuje referenční implementace). K implementaci každého standardního rozšíření se nevyžaduje licence Javy (jako například Apple nebo IBM), ale pokud vyžadována je, mělo by se dodržovat rozhraní standardního rozšíření. V této knize upozorňujeme na jakýkoli kód, který závisí na standardním rozhraní. Malé množství kódu zde, kromě kódu uvedeného v samotné knize, závisí na nestandardních rozšířeních. Můj vlastní balík `com.darwinsys` obsahuje některé obslužné třídy, které se tu a tam používají; jeho import uvidíte v horní části každého souboru, jež používá třídy z tohoto balíku.

Navíc jsou standardizovány další dvě *platformy*, J2ME a J2EE. J2ME (Java 2 Micro Edition) se stará o malá zařízení jako kapesní počítače (PalmOS a jiné), mobilní telefony, faxová zařízení a podobné přístroje. V rámci J2ME existují různé „profily“ pro různé třídy zařízení. J2EE (Java 2 Enterprise Edition) se naopak stará o vytváření velkých, škálovatelných, distribuovaných aplikací. Součástí J2EE jsou API Servlets, JavServer Pages, JavaServer Faces, CORBA, RMI, JavaMail, Enterprise JavaBeans™ (EJB), Transactions a další API. Balíky J2ME a J2EE mají běžně prefix „`javax`“, protože nejsou základními balíčky J2SE. V této knize se nebudeme vůbec zabývat platformou J2ME, nicméně je zde obsaženo několik málo API J2EE, která jsou užitečná na straně klienta, například RMI a JavaMail. Jak jsme se dříve zmínili, pojednání o servletech a JSP z prvního vydání bylo odstraněno, protože nyní již existuje kniha *Servlet and JSP Cookbook*.

Jiné knihy

Do této knihy jsme napěchovali množství užitečných informací. V důsledku rozsahu námětů však není možné poskytnout jakémukoli jednomu tématu pojednání o délce samostatné knihy. Kvůli tomu kniha rovněž obsahuje odkazy na celou řadu webových stránek a dalších knih. To je v souladu s naším cílovým publikem: člověkem, který se chce dozvědět více informací o Javě.

* Všimněte si, že všechny balíky s názvy `javax` představují rozšíření: `javax.swing` a jeho podbalíky (balíky Swing GUI) bývaly rozšířeními, ale nyní jsou v základním API.