Sadržaj

Predgovor ..... **xiii**

1. Početak: Prevođenje i izvršavanje Jave **1**
1.0 Uvod 1
1.1 Prevođenje i izvršavanje Jave: Standardni JDK 2
1.2 Prevođenje i izvršavanje Jave: GraalVM za bolje performanse 3
1.3 Prevođenje, izvršavanje i testiranje sa IDE 5
1.4 Istraživanje Jave sa Jshell-om 11
1.5 Efektivno korišćenje CLASSPATH-a 13
1.6 Preuzimanje i upotreba primera koda 15
1.7 Automatizovane zavisnosti, prevođenja, testiranja i uvođenja alatkom Apache Maven 23
1.8 Automatizovanje zavisnosti, prevođenja, testiranja i uvođenja pomoću alatke Gradle 27
1.9 Upozorenja o zastarelosti 29
1.10 Održavanje ispravnosti koda sa jediničnim testiranjem: JUnit 31
1.11 Održavanje koda pomoću servera Continuous Integration 34
1.12 Dobijanje čitljivih praćenja steka 38
1.13 Pronalaženje još izvornog Java koda 39
1.14 Pronalaženje izvršivih Java biblioteka 40

2. Saradnja sa okruženjem **43**
2.0 Uvod 43
2.1 Uzimanje promenljivih okruženja 43
2.2 Uzimanje informacija iz sistemskih svojstava 44
2.3 Rad sa kodom koji zavisi od verzije Jave ili od operativnog sistema 46
2.4 Upotreba proširenja ili drugih pakovanih API-ja 51
2.5 Upotreba sistema Java modula 52

3. Stringovi i tako to 57
3.0 Uvod 57
3.1 Rastavljanje stringova sa podnizovima ili deljenjem u tokene 60
3.2 Sastavljanje Stringova sa StringBuilderom 65
3.3 Obrada stringa znak po znak 67
3.4 Poravnavanje, uvlačenje, i poništenje uvlačenja za stringove 69
3.5 Konvertovanje između Unicode znakova i stringova 73
3.6 Obrtanje stringa po rečima ili po znakovima 75
3.7 Širenje i skupljanje tabulatora 76
3.8 Kontrola velikih i malih slova 81
3.9 Unošenje znakova koji se ne štampaju 82
3.10 Odsecanje razmaka na krajevima stringa 83
3.11 Kreiranje poruke sa I18N resursima 85
3.12 Upotreba određenog lokalnog standarda 88
3.13 Kreiranje paketa resursa 89
3.14 Program: jednostavan formater teksta 90
3.15 Program: Soundex poređenje imena 92

4. Uparivanje šablona sa regularnim izrazima 97
4.0 Uvod 97
4.1 Sintaksa regularnih izraza 99
4.2 Upotreba regexa u Javi: testiranje šablona 106
4.3 Pronalaženje uparenog teksta 109
4.4 Zamena uparenog teksta 112
4.5 Štampanje svih pojavljivanja šablona 115
4.6 Štampanje redova koji sadrže šablon 118
4.7 Kontrola velikih i malih slova u regularnim izrazima 119
4.8 Uparivanje znakova sa akcentom ili kompozitnih znakova 120
4.9 Uparivanje znaka za novi red u tekstu 121
4.10 Program: raščlanjivanje Apache dnevnika 123
4.11 Program: Potpuni Grep 125

5. Brojevi 131
5.0 Uvod 131
5.1 Provera da li je string valjan broj 134
5.2 Konvertovanje brojeva u objekte i obratno 135
5.3 Uzimanje razlomljenog dela celog broja bez korišćenja pokretnog zareza 136
5.4 Rad sa brojevima u pokretnom zarezu 137
5.5 Formatiranje brojeva 142
5.6 Konvertovanje između Binary, Octal, Decimal i Hexadecimal 146
5.7 Operacije na nizu celih brojeva 147
5.8 Formatiranje sa ispravnom množinom 149

5.9 Generisanje slučajnih brojeva 151
5.10 Množenje matrica 153
5.11 Korišćenje kompleksnih brojeva 155
5.12 Rad sa vrlo velikim brojevima 158
5.13 Program: TempConverter 160
5.14 Program: brojčani palindromi 162

6. Datumi i vremena ... 167
6.0 Uvod 167
6.1 Pronalaženje današnjeg datuma 170
6.2 Formatiranje datuma i vremena 172
6.3 Konvertovanje među datumima/vremenima, YMDHMS i sekundama epohe 174
6.4 Raščlanjivanje stringova u datume 175
6.5 Razlika između dva datuma 176
6.6 Dodavanje na datum ili oduzimanje od njega 177
6.7 Rad sa ponavljajućim događajima 178
6.8 Računanje datuma koji uključuju vremenske zone 181
6.9 Interfejs sa nasleđenim klasama Date i Calendar 182

7. Strukturisanje podataka sa Javom ... 185
7.0 Uvod 185
7.1 Upotreba nizova za strukturisanje podataka 186
7.2 Promena veličine niza 188
7.3 Radni okvir Collections Framework 189
7.4 Kao niz, ali dinamičnije 190
7.5 Upotreba generičkih tipova u vašoj vlastitoj klasi 194
7.6 Kako da iteriram? Da vam nabrojim načine 197
7.7 Izbegavanje duplikata pomoću interfejsa Set 201
7.8 Strukturisanje podataka u Linked List 202
7.9 Mapiranje sa Hashtable i HashMap 207
7.10 Spremanje stringova u svojstva i preference 209
7.11 Sortiranje kolekcije 213
7.12 Izbegavanje želje za sortiranjem 218
7.13 Pronalaženje objekta u kolekciji 220
7.14 Konvertovanje kolekcije u niz 222
7.15 Kako da podaci budu iterabilni 223
7.16 Korišćenje steka objekata 226
7.17 Višedimenzionalne strukture 229
7.18 Uprošćavanje objekata podataka sa Lombok ili Record 231
7.19 Program: poređenja vremena 233

8. Objektno orijentisane tehnike 237
8.0 Uvod 237
8.1 Metodi objekata: formatiranje objekata sa toString(), poređenje sa Equals 240
8.2 Korišćenje unutrašnjih klasa 247
8.3 Obezbeđenje povratnog poziva putem interfejsa 249
8.4 Polimorfizam/Apstraktni metodi 253
8.5 Nabrajanja bezbednog tipa 255
8.6 Izbegavanje NPE izuzetka sa Optional 259
8.7 Nametanje obrasca Singleton 261
8.8 Izbacujte svoje vlastite izuzetke 263
8.9 Upotreba injektovanja zavisnosti 265
8.10 Program: Plotter 268

9. Tehnike funkcionalnog programiranja: funkcionalni interfejsi, tokovi i paralelne kolekcije 273
9.0 Uvod 273
9.1 Lambde/zatvaranja umesto unutrašnjih klasa 276
9.2 Lambda – unapred definisani interfejsi umesto vaših 280
9.3 Uprostite obradu sa Streams 283
9.4 Uprostite Streams sa Collectors 284
9.5 Poboljšanje propusnosti sa paralelnim tokovima i kolekcijama 287
9.6 Upotreba postojećeg koda kao funkcionalnog sa referencama metoda 289
9.7 Java miksini: mešanje u metodima 293

10. Ulaz i izlaz: čitanje, pisanje i trikovi sa direktorijem 297
10.0 Uvod 297
10.1 InputStream/OutputStream i Reader/Writer 299
10.2 Čitanje tekstualnog fajla 301
10.3 Čitanje sa standardnog ulaza ili sa konzole/ kontrolnog terminala 304
10.4 Štampanje sa Formatterom i printf-om 308
10.5 Skeniranje ulaza sa StreamTokenizer-om 312
10.6 Skeniranje ulaza sa klasom Scanner 316
10.7 Skeniranje ulaza sa gramatičkom strukturom 319
10.8 Kopiranje fajla 324
10.9 Ponovno dodeljivanje standardnih tokova 325
10.10 Dupliciranje toka tokom pisanja; Ponovno dodeljivanje standardnih tokova 326
10.11 Čitanje/pisanje drugačijeg skupa znakova 329
10.12 Ti dosadni znakovi za kraj reda 330
10.13 Pazite u fajlu na kôd zavisan od platforme 330
10.14 Čitanje/pisanje binarnih podataka 331
10.15 Čitanje i pisanje JAR ili ZIP arhiva 332

10.16 Pronalaženje fajlova na način nezavisan od sistema fajlova sa getResource() i getResourceAsStream() 336
10.17 Pribavljanje informacija o fajlu: fajlovi i putanje 338
10.18 Kreiranje novog fajla ili direktorijuma 345
10.19 Promena imena fajla ili drugih atributa 346
10.20 Brisanje fajla 349
10.21 Kreiranje prolaznog/privremenog fajla 351
10.22 Listanje direktorijuma 353
10.23 Dobijanje korenih direktorijuma 354
10.24 Upotreba servisa FileWatcher za dobijanje obaveštenja o promenama fajlova 356
10.25 Program: čuvanje korisnikovih podataka na disk 358
10.26 Program: Find – šetnja po stablu fajlova 361

11. Nauka o podacima i R ... 367
11.1 Mašinsko učenje sa Javom 368
11.2 Korišćenje podataka u Apache Spark-u 369
11.3 Koristiti R interaktivno 372
11.4 Poređenje/biranje R implementacija 374
11.5 R iz Java aplikacije: Renjin 375
11.6 Korišćenje Jave iz jedne R sesije 377
11.7 FastR, GraalVM implementacija R-a 378
11.8 Korišćenje R-a u veb aplikaciji 380

12. Mrežni klijenti ... 383
12.0 Uvod 383
12.1 HTTP/REST veb klijent 385
12.2 Uspostavljanje veze sa soket serverom 388
12.3 Pronalaženje i javljanje mrežnih adresa 389
12.4 Obrada mrežnih grešaka 391
12.5 Čitanje i pisanje tekstualnih podataka 392
12.6 Čitanje i pisanje binarnih ili serijalizovanih podataka 394
12.7 UDP Datagrami 397
12.8 URI, URL, ili URN? 400
12.9 Program: TFTP UDP klijent 401
12.10 Program: klijent za ćaskanje zasnovan na soketima 406
12.11 Program: jednostavan proveravač HTTP linkova 410

13. Java na serverskoj strani ... 413
13.0 Uvod 413
13.1 Otvaranje serverskog soketa za rad 414
13.2 Pronalaženje mrežnih interfejsa 417

13.3 Vraćanje odgovora (string ili binary) 418
13.4 Vraćanje informacija o objektu preko mrežne veze 422
13.5 Rad sa više klijenata 423
13.6 Nuđenje protokola HTTP 428
13.7 Obezbeđivanje veb servera sa SSL i JSSE 430
13.8 Kreiranje REST servisa sa JAX-RS-om 433
13.9 Mrežno evidentiranje 436
13.10 Postavljanje SLF4J-a 437
13.11 Mrežno evidentiranje Log4j 439
13.12 Mrežno evidentiranje pomoću java.util.logging-a 444

14. Obrada JSON podataka ... 449
14.0 Uvod 449
14.1 Direktno generisanje JSON-a 451
14.2 Raščlanjivanje i pisanje JSON-a - Jackson 452
14.3 Raščlanjivanje i pisanje JSON-a – org.json 453
14.4 Raščlanjivanje i pisanje JSON-a – JSON-B 455
14.5 Pronalaženje JSON elemenata – JSON Pointer 457

15. Paketi i pakovanje ... 461
15.0 Uvod 461
15.1 Pravljenje paketa 462
15.2 Dokumentovanje klasa – Javadoc 464
15.3 Dalje od Javadoca: anotacije/metapodaci 468
15.4 Priprema klase da bude JavaBean 469
15.5 Arhiviranje alatkom JAR 473
15.6 Izvršavanje programa iz JAR-a 474
15.7 Pakovanje komponenti za Veb sloj u WAR fajl 476
15.8 Kreiranje manje distribucije – jlink 478
15.9 Upotreba JPMS-a za kreiranje modula 479

16. Java sa nitima ... 483
16.0 Uvod 483
16.1 Izvršavanje koda u drugoj niti 485
16.2 Prikaz pokretne slike sa animacijom 489
16.3 Zaustavljanje niti 494
16.4 Sastanci i tajm-autevi 497
16.5 Sinhronizovanje niti pomoću ključne reči synchronized 498
16.6 Jednostavnija sinhronizacija pomoću brava 504
16.7 Jednostavniji proizvođač/potrošač sa interfejsom Queue 508
16.8 Optimizacija paralelne obrade sa Fork/Join 511
16.9 Raspoređivanje zadataka: buduća vremena, pozadinsko snimanje u editoru 514

17. Refleksija, ili „Klasa po imenu Class" **519**
17.0 Uvod 519
17.1 Pribavljanje deskriptora klase 520
17.2 Pronalaženje i korišćenje metoda i polja 521
17.3 Pristup privatnim metodima i poljima – Reflection 525
17.4 Dinamičko učitavanje i instanciranje klase 526
17.5 Konstruisanje klase od nule koristeći ClassLoader 529
17.6 Konstruisanje klase od nule koristeći JavaCompiler 530
17.7 Merenje performansi 533
17.8 Štampanje informacija o klasi 537
17.9 Lista klasa u paketu 539
17.10 Upotreba i definisanje anotacija 542
17.11 Pronalaženje klasa sličnih dodatnim programima pomoću anotacija 547
17.12 Program: CrossRef 549

18. Java sa drugim jezicima **555**
18.0 Uvod 555
18.1 Izvršavanje spoljnog programa iz Jave 556
18.2 Izvršavanje programa i hvatanje njegovog izlaza 560
18.3 Pozivanje drugih jezika – javax.script 564
18.4 Mešanje jezika – GraalVM 566
18.5 Povezivanje Jave i Perla 567
18.6 Pozivanje drugih jezika putem matičnog koda 571
18.7 Pozivanje Jave iz matičnog koda 576

Pogovor **579**

Java nekad i sad **581**

Indeks **591**

Predgovor

Kao bilo koji od najčešće korišćenih programskih jezika, Java ima svoje kritičare, branioce, probleme, smicalice [1] i krivu učenja. *Java kuvar* ima za cilj da pomogne Java programeru da ubrzano savlada neke od najvažnijih delova Java programiranja. Usredsređujem se na standardne API-je i neke API-je drugih dobavljača, ali se ne ustručavam da obradim i probleme jezika.

Ovo je četvrto izdanje ove knjige i oblikovali su je mnogi ljudi i bezbrojne promene koje je Java pretrpela tokom prve dve decenije svoje popularnosti. Čitaoci koje zanima istorijat Jave mogu ga naći u Dodatku A.

Java 11 trenutno nije dugoročno podržana verzija, ali su izašle i Java 12 i 13. Java 14 je u fazi ranog pristupa i njeno konačno izdanje je zakazano za isti dan kao i četvrto izdanje ove knjige. Novi ritam izdanja svakih šest meseci je možda odličan za razvojni tim Java SE u Oraclu i za sajtove novosti vezanih za Javu kojima se pristupa jednim klikom, ali za autore knjiga o Javi „može da znači dodatni posao", pošto knjige obično imaju duži ciklus revizija nego što ga Java sada ima! Java 9, koja je izašla nakon prethodnog izdanja ove knjige, bila je prelomno izdanje, prvo izdanje koje je posle dugog vremena prekinulo kompatibilnost unazad, pre svega sistemom Java modula. Za sve što se nalazi u knjizi se pretpostavlja da će raditi na svakoj JVM koja se još uvek koristi za razvijanje koda. Niko ne bi smeo da koristi Javu 7 (ili nešto još starije!) za bilo šta, a niko ne bi smeo da piše nešto novo u Javi 8. Ako to još uvek radite, vreme je da krenete dalje!

Cilj ove revizije je da se knjiga dovede u sklad sa svim ovim promenama. Dok sam dosta starijeg materijala izbacio, dodao sam informacije o novim funkcijama kao što su Moduli i interaktivni JShell, a usput sam ažurirao mnoge druge informacije.

[1] Za smicalice, pogledajte knjige *Java Puzzlers*. Autori: Joshua Bloch i Neal Gafter (Addison-Wesley).

Za koga je ova knjiga

Poći ću od pretpostavke da su vam poznate osnove Jave. I neću da vam objašnjavam kako da uradite `println` stringa, niti kako da napišete klasu koja proširuje drugu klasu, ili implementira neki interfejs. Pretpostavljam da ste pohađali neki kurs Jave kao što je *Learning Tree – Uvod* ili da ste proradili neku uvodnu knjigu kao što je *Java programiranje bez oklevanja*, *Učenje Jave*, ili *Java ukratko* (O'Reilly). međutim, Poglavlje 1 pokriva neke tehnike koje možda ne znate najbolje, a neophodne su da bi se razumelo nešto u kasnijem materijalu. Slobodno preskačite! I štampana verzija knjige i digitalno izdanje imaju ozbiljne unakrsne reference.

Šta ima u ovoj knjizi?

Činilo se da Java bolje odgovara za „programiranje naveliko", ili programiranje aplikacija za preduzeća, nego za jednoredno skriptovanje kao u jezicima Perl, Awk, ili Python. To je zato što je to prevedeni, objektno orijentisan jezik. Međutim, ta prilagođenost se donekle promenila pojavom JShell-a (videti Recept 1.4). Mnoge tehnike ilustrujem kraćim primerima Java klasa, pa čak i fragmentima koda; neki od jednostavnijih će biti prikazani pomoću JShell-a. Svi primeri koda (osim nekih od po jedan ili dva reda) nalaze se u nekom od mojih javnih GitHub spremišta, pa možete biti uvereni da je svaki fragment koda koji vidite ovde već bio preveden i da je većina nedavno izvršavana.

Neki od dužih primera u ovoj knjizi su alatke koje sam prvobitno pravio za automatizovanje nekog svakodnevnog posla. Na primer, alatka po imenu `MkIndex` (u spremištu *javasrc*) učitava direktorijum najvišeg nivoa na mestu gde čuvam izvorni kôd mojih Java primera i pravi od tog direktorijuma fajl *index.html* pogodan za pretraživač. Još jedan primer je `XmlForm`, koji se koristio za konvertovanje delova rukopisa iz XML-a u oblik potreban za ostali izdavački softver. `XmlForm` je takođe izvršio – uz pomoć još jednog programa, `GetMark` – ubacivanje celih i delimičnih delova koda iz direktorijuma *javasrc* u rukopis knjige. `XmlForm` je uključen u pomenuto spremište Github, a takođe i jedna kasnija verzija programa `GetMark`, mada nijedan od njih nije korišćen za izradu četvrtog izdanja. Ovih dana, O'Reilly-jev izdavački softver Atlas koristi Asciidoctor, koji obezbeđuje mehanizam koji koristimo za ubacivanje fajlova i delova fajlova u knjigu.

Organizacija ove knjige

Pregledajmo organizaciju ove knjige. Svako poglavlje se sastoji od više recepata, kratkih odeljaka koji opisuju neki problem i njegovo rešenje, zajedno sa primerom koda. Kôd u svakom receptu je zamišljen da bude uglavnom samostalan; slobodno upotrebite delove iz njih u svojim projektima. Kôd se distribuira sa autorskim pravima u stilu Berkeley, samo da bi se obeshrabrilo reprodukovanje na veliko.

Poglavlje 1, Početak: Prevođenje i izvršavanje Jave, započinjem opisom nekih metoda za prevođenje vašeg programa na različitim platformama, njihovog izvršavanja u

različitim okruženjima (pretraživač, komandna linija, radna površina sa prozorima), i za istraživanje grešaka.

Poglavlje 2, Saradnja sa okruženjem, prelazi od prevođenja i izvršavanja programa na njegovo prilagođavanje okolini – ostalim programima u vašem računaru.

Sledećih nekoliko poglavlja bavi se osnovnim API-jima. *Poglavlje 3, Stringovi i tako to*, koncentriše se na jedan od najosnovnijih ali moćnih tipova podataka u Javi, pokazuje vam kako da sastavljate, secirate, poredite i preuređujete ono što biste inače smatrali običnim tekstom. Ovo poglavlje takođe pokriva temu internacionalizacije/lokalizacije tako da vaši programi mogu da rade u Akbaru, Afganistanu, Alžiru, Amsterdamu i Engleskoj kao što rade u Alberti, Arkanzasu i Alabami.

Poglavlje 4, Uparivanje šablona sa regularnim izrazima, uči vas kako da koristite moćnu tehnologiju regularnih izraza iz Unixa u mnogim problemskim domenima uparivanja stringova i šablona. Regex obrada je već godinama standard u Javi, ali ako ne znate kako da je koristite, možda ćete ponovo izmišljati toplu vodu.

Poglavlje 5, Brojevi, bavi se kako ugrađenim numeričkim tipovima kao što su `int` i `double`, tako i odgovarajućim API klasama (`Integer`, `Double`, itd.) i mogućnostima za konverzije i ispitivanja koje one nude. Takođe se ukratko pominju klase za rad sa „velikim brojevima" (engl.*„big number classes"*). Pošto Java programeri često moraju da se bave datumima i vremenima, kako lokalnim tako i međunarodnim, *Poglavlje 6, Datumi i vremena*, pokriva ovu važnu temu.

Sledećih nekoliko poglavlja pokriva obradu podataka. Kao u većini jezika, nizovi su u Javi linearne, indeksirane kolekcije sličnih objekata, kao što se opisuje u *Poglavlju 7, Strukturisanje podataka sa Javom*. Ovo poglavlje se nadalje bavi mnogim klasama Collections: moćnim načinom za čuvanje velike količine objekata u paketu `java.util`, uključujući korišćenje Java Generics-a.

Uprkos izvesnoj sintaktičkoj sličnosti sa proceduralnim jezicima kao što je C, Java je u biti Objektno Orijentisan Programski (OOP) jezik, u koji su vešto ugrađeni neki važni konstrukti Funkcionalnog Programiranja (FP). *Poglavlje 8, Objektno orijentisane tehnike*, opisuje neke od ključnih pojmova OOP-a, kako se odnose na Javu, uključujući uobičajeno redefinisane metode iz `java.lang.Object` i važno pitanje projektnih obrazaca. Java nije, i nikad neće biti, čist FP (engl. *Functional Programming*) jezik. Međutim, moguće je koristiti neke aspekte FP-a, pogotovo sa Javom 8 i njenom podrškom za lambda izraze (tzv. zatvaranja – *closures*). To se opisuje u *Poglavlju 9, Tehnike funkcionalnog programiranja: funkcionalni interfejsi, tokovi i paralelne kolekcije*.

Naredno poglavlje se bavi aspektima tradicionalnog ulaza i izlaza. *Poglavlje 10. Ulaz i izlaz: čitanje, pisanje i trikovi sa direktorijem*, sadrži detaljna pravila za čitanje i pisanje fajlova (nemojte ovo preskočiti ako mislite da su fajlovi dosadni; neke od ovih informacija će vam biti potrebne u kasnijim poglavljima). Ovo poglavlje vam prikazuje i sve ostalo o fajlovima – kao što je utvrđivanje njihove veličine i vremena kada su poslednji

put menjani – i o čitanju i menjanju direktorijuma, kreiranju privremenih fajlova i preimenovanju fajlova na disku.

Big Data i nauka o podacima su postali važni, a Java je upravo uključena u to. Apache Hadoop, Apache Spark, i još mnogo toga u infrastrukturi Big Data napisano je u Javi i može se proširiti u Javi, kao što je opisano u *Poglavlju 11, Nauka o podacima i R*. Programski jezik R je popularan među naučnicima koji se bave podacima, statističarima i drugim naučnicima. Postoje bar dve implementacije R-a kodirane u Javi, a Java može i direktno da se poveže sa standardnom R implementacijom u oba smera, zato ovo poglavlje pokriva takođe i R.

Pošto je Java prvobitno proglašena programskim jezikom za internet, red je da se neko vreme posveti umrežavanju u Javi. *Poglavlje 12, Mrežni klijenti*, pokriva osnove mrežnog programiranja za klijentsku stranu, sa fokusom na sokete. Danas je tolikim aplikacijama potreban pristup veb servisima, pre svega RESTful veb servisima, da je to izgledalo neophodno. Zatim prelazim na serversku stranu u *Poglavlju 13, Java na serverskoj strani*, gde ćete naučiti neke tehnike programiranja za serversku stranu.

Jedno jednostavno tekstualno predstavljanje za razmenu podataka je JSON, JavaScript notacija objekata. *Poglavlje 14, Obrada JSON podataka*, opisuje format i neke od mnogih API-ja koji su se pojavili u tu svrhu.

Poglavlje 15, Paketi i pakovanje, pokazuje kako se prave pakovanja klasa koje sarađuju. U ovom poglavlju je takođe reč o uvođenju (tzv. distribuciji i instaliranju) vašeg softvera.

Poglavlje 16, Java sa nitima, vas uči kako da pišete klase koje naizgled rade više stvari u isto vreme i daju vam mogućnost da iskoristite moćan višeprocesorski hardver.

Poglavlje 17, Refleksija, ili „Klasa po imenu Class", otkriva vam tajne kao što su mehaničko pisanje unakrsno referenciranih API dokumenata i kako su veb serveri u stanju da učitaju bilo koji Servlet – iako nikad ranije nisu videli tu konkretnu klasu – i izvrše ga.

Ponekad već imate kôd koji je napisan i funkcioniše u nekom drugom jeziku i može da vam obavi deo posla, ili hoćete da upotrebite Javu kao deo većeg paketa. *Poglavlje 18, Java sa drugim jezicima*, pokazuje kako se izvršava spoljni program (preveden ili kao skript) i takođe direktno sarađuje sa matičnim kodom u jezicima C/C++ ili drugim.

U knjizi ovih dimenzija nema mesta za sve što bih voleo da vam kažem o Javi. Pogovor izlaže neka završna razmišljanja i vezu do mog kratkog onlajn pregleda Java API-ja za koje bi svaki Java programer morao da zna.

Na kraju, *Dodatak A*, *Java nekad i sad*, je priča o istoriji Jave, izdanje po izdanje, tako da, bez obzira na to koju verziju Jave ste naučili, možete da uskočite ovde i brzo postignete ažurnost.

Toliko tema, a tako malo stranica! Mnoge teme ne dobijaju 100% pokrivenosti; pokušao sam da uključim najvažnije delove svakog API-ja. Da biste pošli dalje, otvorite zvanične *javadoc* stranice svakog paketa; mnoge od njih sadrže neke kratke informacije za obuku o načinu korišćenja tog paketa.

Pored delova Jave pokrivenih u ovoj knjizi, standardizovana su izdanja za još dve platforme, Java ME i Java EE. Java Micro Edition (Java ME) se odnosi na male uređaje kao što su ručni računari, mobilni telefoni i faks mašine. Na drugom kraju skale – velike serverske mašine – tu je Eclipse Jakarta EE, zamena za raniji Java EE, koji je u prethodnom veku bio poznat kao J2EE. Jakarta EE se odnosi na izgradnju velikih, skalabilnih, distribuiranih aplikacija. API-ji koji spadaju u Jakarta EE uključuju Servlets, JavaServer Pages, JavaServer Faces, JavaMail, Enterprise JavaBeans (EJBs), CDI (Container and Dependency Injection) i Transactions. Jakarta EE paketi obično počinju sa „javax" jer to nisu osnovni paketi. U ovoj knjizi se pominje samo malo njih; tu je takođe *Java EE 8 Cookbook* Eldera Moraesa (O'Reilly) koji pokriva neke API-je Jakarte EE, kao i jedan stariji *Java Servlet & JSP Cookbook* Brucea Perryja (O'Reilly).

Ova knjiga ne pokriva Javu ME (Java Micro Edition). Ni malo. Ali kad je reč o mobilnim telefonima i drugim mobilnim uređajima, verovatno znate da Android koristi Javu kao svoj jezik. Java programere bi trebalo da teši to da Android takođe koristi većinu osnovnih Java API-ja, ali ne i Swing i AWT, za koje ima zamene specifične za Android. Java programer koji želi da uči Android može da uzme moju knjigu *Android Cookbook* (O'Reilly), ili veb sajt te knjige.

Java knjige

U ovu knjigu je upakovano mnogo korisnih informacija. Međutim, zbog obimnosti tema, nije moguće svakoj od njih posvetiti pažnju kao što bi to bio slučaj u celoj knjizi. Zbog toga, knjiga sadrži reference na mnoge veb sajtove i druge knjige. Nadam se da isticanjem tih referenci zadovoljavam svoju ciljnu publiku: one koji žele da nauče više o Javi.

O'Reilly izdaje, po mom mišljenju, najbolji izbor Java knjiga na tržištu. Kako API nastavlja da se širi, širi se i pokrivanje. Pregledajte kompletnu listu O'Reilly kolekciju Java knjiga; možete da ih kupite u većini knjižara, kako fizičkih tako i virtuelnih. Možete takođe da ih čitate onlajn preko O'Reilly Online Learning Platform, servisa za koji se plaća pretplata. I, naravno, većina je sada dostupna u formatu e-knjige; e-knjige O'Reilly su slobodne od DRM (Digital Rights Management) zaštite, pa ne morate da brinete da će njihova šema zaštite da vas ograniči na konkretan uređaj ili sistem, kao kod nekih drugih izdavača.

Mada se na odgovarajućim mestima u knjizi pominju mnoge knjige, neke od njih zaslužuju da ih ovde posebno istaknemo.

Prvo i pre svega, *Java ukratko* Davida Flanagana i Benjamina Evana (O'Reilly) nudi kratak pregled jezika i API-ja uz detaljne reference na najbitnije pakete. To je praktično da se drži pored računara. *Java bez oklevanja* Berta Batesa i Kathy Sierre nudi daleko zabavniji uvod u jezik, pa se preporučuje manje iskusnim programerima.

Java 8 Lambdas (Warburton, O'Reilly) pokriva Lambda sintaksu uvedenu u Javi 8 za podršku funkcionalnom programiranju i uopšte konciznijem kodu.

Java 9 Modularity: Patterns and Practices for Developing Maintainable Applications – autora Sander Mak i Paul Bakker (O'Reilly), pokriva značajne izmene jezika u Javi 9 za Java sistem modula.

Java Virtual Machine Jona Meyera i Troya Downinga (O'Reilly) će privući nekoga ko želi da sazna više o onome šta je ispod haube. Ova knjiga je rasprodata, ali može se naći korišćena i u bibliotekama.

Konačan (i monumentalan) opis programiranja Swing GUI-a je *Java Swing* Roberta Ecksteina i drugih (O'Reilly).

Java Network Programming i *Java I/O*, oba Elliotte Harold (O'Reilly), su takođe korisne reference.

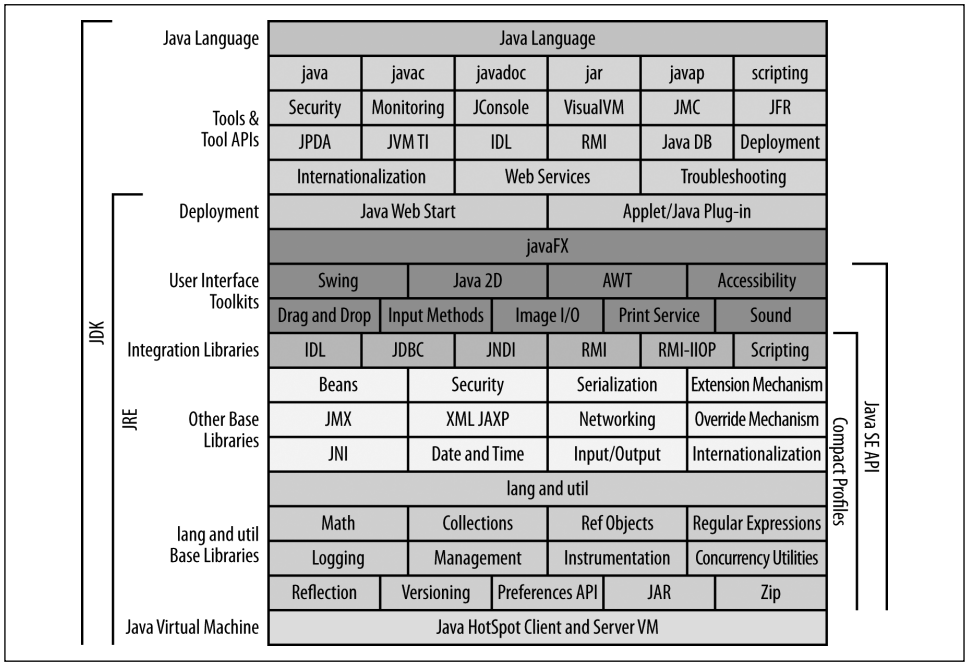
Za rad na Java bazama podataka preporučuju se *Database Programming with JDBC & Java* Georgea Reesea (O'Reilly) i *Pro JPA 2: Mastering the Java Persistence API* Mikea Keitha i Merricka Schincariola (Apress). Ili moj predstojeći overview of Java Database.

Mada knjiga koju trenutno čitate ne pokriva mnogo Java EE, voleo bih da pomenem dve knjige o toj temi:

- Arun Gupta pokriva izdanje Enterprise Edition u *Java EE 7 Essentials* (O'Reilly).
- Adam Bien-ov *Real World Java EE Patterns: Rethinking Best Practices* nudi korisne uvide u dizajn i implementaciju Enterprise aplikacija.

Na veb sajtu O'Reilly možete ih naći još.

Konačno, mada to nije knjiga, Oracle ima mnogo Java informacija na Vebu. Ta veb stranica je imala jedan veliki dijagram sa svim komponentama Jave u jednom „konceptualnom dijagramu". Jedna rana verzija tog dijagrama je prikazana na Slici P-1; tamo je svaki obojeni okvir predstavljao vezu koja reaguje na pritisak mišem ka detaljima te konkretne tehnologije.



Slika P-1. Konceptualni dijagram Jave iz Oracle dokumentacije

Kako bilo, novije verzije Jave zamenile su ovo tekstualnom stranicom; za Javu 13 stranica se nalazi na adresi *https://docs.oracle.com/en/java/javase/13*.

Opšte knjige o programiranju

Knjiga Donalda E. Knutha *Umetnost programiranja na računaru* (Addison-Wesley) je bila izvor nadahnuća za generacije studenata računarstva od kad je prvi put objavljena 1968. godine. Prva knjiga pokriva *Fundamentalne algoritme*, druga knjiga je *Polu-numerički algoritmi*, treća knjiga je *Sortiranje i pretraživanje*, a knjiga 4A je *Kombinatorijski algoritmi, Deo 1*. Preostale knjige u projektovanoj seriji nisu dovršene. Mada su njegovi primeri daleko od Jave (on je za svoje primere izmislio hipotetički asemblerski jezik MIX), mnogi od njegovih opisa algoritama – o tome kako bi računare trebalo koristiti za rešavanje realnih problema – relevantni su i danas kao što su bili pre toliko godina.[2]

[2] Uz moguće izuzetke za odluke algoritama koje su danas manje relevantne zbog ogromnih promena u sada dostupnoj moći računara.

Mada su njeni primeri koda već prilično zastareli, knjiga *The Elements of Programming Style* – autora Briana Kernighana i P. J. Plaugera (McGraw-Hill) određuje stil (bukvalno) za generaciju programera sa primerima iz različitih strukturisanih programskih jezika. Kernighan i Plauger su napisali još dve knjige, *Software Tools* (Addison-Wesley) i *Software Tools in Pascal* (Addison-Wesley), koje daju toliko dobrih saveta o programiranju da sam svim programerima savetovao da ih čitaju. Međutim, ove tri knjige su sada zastarele; mnogo puta sam poželeo da napišem knjigu koja bi bila njihov nastavak u nekom modernijem jeziku. Umesto toga sada upućujem na *The Practice of Programming*, Kernighanov nastavak – koji je pisao sa Robom Pikom (Addison-Wesley) – nastavak serije *Software Tools*. Ta knjiga nastavlja tradiciju izvrsnih udžbenika softvera iz laboratorija Bell Labs. U prethodnim izdanjima ove moje knjige, čak sam prilagodio jedan deo koda iz njihove knjige, njihov CSV raščlanjivač. Konačno, Kernighan je nedavno objavio *UNIX: A History and a Memoir*, njegov pogled na istoriju Unixa.

Vidite takođe *The Pragmatic Programmer* – autora Andrewa Hunta i Davida Thomasa (Addison-Wesley).

Knjige o dizajnu

Java Design – autor Peter Coad (PTR-PH/Yourdon Press) opisuje pitanja objektno orijentisane analize i dizajna, konkretno za Javu. Coad je donekle kritičan prema Javinoj implementaciji paradigme observable-observer i nudi svoju vlastitu zamenu.

Jedna od najčuvenijih knjiga o objektno orijentisanom dizajnu poslednjih godina je *Projektni obrasci* (*Design Patterns*) – autori Erich Gamma, Richard Helm, Ralph Johnson i John Vlissides (Addison-Wesley). Ove autore često kolektivno zovu „Gang of Four," pa zato njihovu knjigu ponekad zovu „GoF knjiga". Jedan od mojih kolega je rekao da je to „najbolja ikad napisana knjiga o objektno orijentisanom dizajnu" i slažem se; u najmanju ruku, spada među najbolje.

Refactoring – autor Martin Fowler (Addison-Wesley) pokriva mnoga „pospremanja koda" koja mogu da se primene na kôd da bi se poboljšala čitljivost i održivost. Isto kao što je GoF knjiga uvela novu terminologiju koja pomaže programerima i ostalima da raspravljaju o tome kako treba projektovati kôd, Fowlerova knjiga je uvela rečnik za rasprave o njegovom unapređivanju. Ali ova knjiga je možda manje korisna od drugih; mnoga prepravljanja koda se sada pojavljuju u meniju Refactoring u IDE-u Eclipse (videti Recept 1.3).

Trenutno su u opticaju dva važna toka teorije metodologije. Prvi je poznat kao Agilni metodi, a njegovi najpoznatiji članovi su Scrum i XP (Extreme Programming). XP (metodologija, a ne onaj zaista stari stil Microsoftovog OS-a) predstavljen je u nizu malih, kratkih, čitljivih tekstova koje vodi njihov dizajner, Kent Beck. Prva knjiga u seriji XP je *Extreme Programming Explained* (Addison-Wesley). Dobar pregled svih Agilnih metoda je *Agile Software Development Ecosystems* – autor Jim Highsmith (Addison-Wesley).

Još jedna grupa važnih knjiga o metodologiji, koje obrađuju tradicionalniji objektno orijentisan dizajn, je serija UML koju vode „Tri Amigosa" (Booch, Jacobson i Rumbaugh). Njihova glavna dela su *UML User Guide*, *UML Process* i još neka. Jedna manja i pristupačnija knjiga iz iste serije je *UML Distilled* Martina Fowlera.

Konvencije koje se koriste u ovoj knjizi

U ovoj knjizi se koriste sledeće konvencije.

Konvencije programiranja

U ovoj knjizi koristim sledeću terminologiju. Program je svaka jedinica koda koja može da se izvrši: od glavnog programa sa pet redova, do servleta ili komponente za nivo veba, EJB-a, ili potpuno razrađenih GUI aplikacija. Apleti su bili Java programi za upotrebu u veb pretraživačima; bili su neko vreme popularni, ali danas jedva da postoje. Servlet je Java komponenta izgrađena pomoću Jakarta EE API-ja za upotrebu u veb serveru, obično putem HTTP-a. EJB-ji su komponente poslovnog nivoa izgrađene pomoću Jakarta API-ja. Aplikacija je svaka druga vrsta programa. Stona aplikacija (tzv. klijent) komunicira sa korisnikom. Serverski program se bavi klijentom indirektno, obično putem mrežne veze (ovih dana obično HTTP/HTTPS).

Prikazani primeri su dve vrste. Oni koji počinju sa nula ili više važnih iskaza, jednim javadoc komentarom, i iskazom `public class` su kompletni primeri. Oni koji počinju deklaracijom ili izvršnim iskazom su, naravno, odlomci. Međutim, potpune verzije iz kojih ti odlomci potiču su bile prevedene i izvršene i nalaze se u onlajn izvoru.

Recepti su numerisani po poglavlju i broju, pa se, na primer, Recept 8.1 odnosi na prvi recept u Poglavlju 8.

Konvencije sloga

U ovoj knjizi se koriste sledeće tipografske konvencije:

Kurziv
: Koristi se za komande, imena fajlova i URL-ove primera. Takođe se koristi za naglašavanje i za definisanje novih pojmova kada se prvi put pojave u tekstu.

`Konstantna širina`
: Koristi se u primerima koda kada se prikazuju delimični ili kompletni programski listinzi izvornog Java koda. Koristi se takođe za imena klasa, metoda, promenljivih i za druge fragmente Java koda.

`Zacrnjena konstantna širina`
: Koristi se za korisnički ulaz, kao što su komande koje kucate na komandnoj liniji.

`Konstantna širina kurzivom`
: Pokazuje tekst koji treba zameniti korisnikovim vrednostima ili vrednostima koje zavise od konteksta.



Ovaj element znači savet ili predlog.



Ovaj element znači opštu napomenu.



Ova ikona znači upozorenje ili oprez.

Primeri koda

Uzorci koda za ovu knjigu se nalaze na autorovom GitHub-u. Većina je u spremištu *javasrc*, ali je nekoliko izvučeno iz jednog drugog spremišta, *darwinsys-api*. Detalji za njihovo preuzimanje su u Receptu 1.6.

Mnoge programe prati primer koji ih prikazuje na delu, kada se izvrše sa komandne linije. Obično se prikazuje odzivnik koji se završava sa `$` za Unix ili sa `>` za Windows, zavisno od toga koji sam računar koristio onog dana kad sam pisao taj primer. Ako ispred tog odzivnika stoji neki tekst, on može da se zanemari. To može da bude ime putanje ili ime računara, opet zavisno od sistema.

U ovim primerima će se takođe videti puno ime paketa klase zato što to Java zahteva kada se program pokreće sa komandne linije. A pošto će vas to podsetiti na poddirektorijum izvornog spremišta u kojem se nalazi izvorni kôd, ja neću vrlo često eksplicitno da ukazujem na to.

Mi cenimo, ali uglavnom ne zahtevamo, pripisivanje. Pripisivanje obično sadrži naslov, autora, izdavača i ISBN. Na primer: „*Java Cookbook* by Ian F. Darwin (O'Reilly). Copyright 2020 RejmiNet Group, Inc., 978-1-492-07258-4."

Ako smatrate da vaše korišćenje primera koda prelazi granice fer upotrebe ili gore date dozvole, slobodno nas kontaktirajte na *permissions@oreilly.com*.

O'Reilly Onlajn učenje

Već preko 40 godina, *O'Reilly Media* pruža tehnološku i poslovnu obuku, znanje i uvid kao pomoć kompanijama na putu do uspeha.

Naša jedinstvena mreža eksperata i inovatora deli svoje znanje i stručnost preko knjiga, članaka, konferencija i naše platforme za onlajn učenje. O'Reilly-jeva platforma za onlajn učenje daje vam pristup na zahtev do kurseva obuke uživo, detaljnim putevima učenja, interaktivnim okruženjima za kodiranje i velikoj zbirci tekstova i videa od O'Reilly-ja i više od 200 drugih izdavača. Za više informacija, molim posetite *http://oreilly.com*.

Komentari i pitanja

Kao što je već pomenuto, testirao sam sav kôd najmanje na jednoj od referentnih platformi, a većinu na nekoliko. Ipak, mogu da postoje zavisnosti od platforme, ili čak greške u mom kodu ili u nekoj važnoj implementaciji Jave. Molim da prijavite greške koje pronađete, kao i vaše predloge za buduća izdanja, tako da pišete na:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (u SAD i Kanadi)
707-829-0515 (međunarodno ili lokalno)
707-829-0104 (fax)

Postoji jedna veb stranica za ovu knjigu na kojoj se nalaze greške, primeri i neke dodatne informacije. Može joj se pristupiti na *http://shop.oreilly.com/product/0636920304371.do*.

E-mail *bookquestions@oreilly.com* za komentare i tehnička pitanja o ovoj knjizi.

Za više informacija o našim knjigama, kursevima, konferencijama i novostima vidite naš veb sajt na http://www.oreilly.com.

Nađite nas na Facebooku: *http://facebook.com/oreilly*

Pratite nas na Twitteru: *http://twitter.com/oreillymedia*

Gledajte nas na YouTubu: *http://www.youtube.com/oreillymedia*

Sajt O'Reilly navodi greške pronađene u knjizi. Tu ćete naći i sav izvorni kôd svih primera Java koda za preuzimanje; *molim* nemojte gubiti vreme da ih ponovo kucate sami! Za konkretna uputstva, vidite Recept 1.6.

Priznanja

Napisao sam u Pogovoru prvog izdanja da je „pisanje ove knjige bilo jedno ponižavajuće iskustvo." Trebalo bi da dodam da je i održavanje njene ažurnosti takođe bilo ponižavajuće. Mada su mnogi bili izdašni u svojim pohvalama – jedan vrlo ljubazan recenzent je nazvao „verovatno najboljom knjigom koja je ikada napisana o programskom jeziku Java" – bio sam ponižen brojem grešaka i propusta u ranijim izdanjima. Nastojao sam da ih ispravim.

Moj život je više puta dodirnuo tok sudbine dovodeći me u kontakt sa pravom osobom da mi pokaže pravu stvar u pravo vreme. Steve Munro, sa kojim sam već odavno izgubio vezu, me je upoznao sa računarima kada smo bili u istom razredu u srednjoj školi – konkretno jednim IBM-om 360/30 u Odboru za obrazovanje u Torontu, koji je bio veći od dnevne sobe, imao 32 ili 64*K* (a ne *M* ili *G*!) memorije, i možda imao snagu jednog PC/XT-a. Pokojni Herb Kugel me je uzeo pod svoje na Univerzitetu Toronto dok sam učio o većim IBM mejnfrejmovima koji su došli posle. Terry Wood i Dennis Smith na Univerzitetu Toronto upoznali su me sa mini- i mikro-računarima pre nego što je postojao IBM PC. Uveče i vikendima, Toronto Business Club of Toastmasters International i škola SCUBA Canada Ala Lamberta dozvolili su mi da razvijam svoje sposobnosti javnog govora i predavanja. Nekoliko ljudi na Univerzitetu Toronto, ali posebno Geoffrey Collyer, naučili su me karakteristikama i prednostima operativnog sistema Unix u vreme kada sam bio spreman da ga naučim.

Zahvaljujući mnogim instruktorima i studentima u kompaniji za obuku Learning Tree koji su mi pokazali kako da unapredim svoje prezentacije. Još uvek predajem za „The Tree" i preporučujem njihove kurseve za zauzetog programera koji želi da se detaljno usredsredi na jednu temu tokom četiri dana.

Bliže ovom projektu, Tim O'Reilly je verovao u moju „knjižicu" dok je još bila uzorak poglavlja iz predloženog većeg rada, i omogućio moj rani ulazak u proređen krug O'Reilly autora. Godinama kasnije, Mike Loukides me je ohrabrio da nastavim sa pokušajima da nađem ideju o Java knjizi sa kojom bismo mogli obojica da radimo. I ostao je uz mene kad sam stalno kasnio za rokovima. Mike je takođe pročitao ceo rukopis i dao mnoge razumne komentare, od kojih su neki spustili fantaziranja na zemlju. Jessamyn Read je mnoge škrabotine sumnjive čitljivosti koje je primila faksom ili e-poštom pretvorila u kvalitetne ilustracije koje vidite u ovoj knjizi. I mnogi, mnogi drugi talentovani ljudi u O'Reillyju pomogli su da se ova knjiga dovede u oblik u kojem je sada pred vama.

Primeri koda se sada uključuju dinamički (pa ažuriranje ide brže) umesto da se prenose. Moj sin (i programer funkcionalnog programiranja) Benjamin Darwin pomogao je da se održi rok tako što je skoro celu kodnu bazu konvertovao u O'Reilly-ev najnoviji mehanizam „include" i razrešio nekoliko drugih pitanja prezentacije koja se ne odnose na Javu. Takođe je pomogao da Poglavlje 9 bude jasnije i funkcionalnije.

O'Reilly

Za četvrto izdanje knjige, Suzanne McQuade je bila urednički nadzornik, a Corbin Collins glavni urednik. Corbin je bio posebno pedantan u pregledanju rukopisa. Meghan Blanchette, Sarah Schneider, Adam Witwer, Melanie Yarbrough, i mnogi ljudi u proizvodnji navedeni na stranici autorskih prava su svi imali ulogu u pripremi trećeg izdanja za vaše čitanje. Hvala Mikeu Loukidesu, Deb Cameron i Marlowe Shaeffer za urednički i produkcijski rad na drugom izdanju.

Tehnički recenzenti

Za četvrto izdanje sam imao sreću da dobijem dva vrlo temeljita tehnička recenzenta, Sandera Maka i Daniela Hinojosu. Mnoga pitanja koja nisam bio razmatrao tokom glavne revizije pokrenula su njih dvojica, što je dovelo do opsežnih prepravki i promena u poslednjih nekoliko nedelja pre nego što je O'Reillyjev tim za proizvodnju preuzeo sve. Mnogo hvala obojici!

Moj recenzent za treće izdanje, Alex Stangl, pročitao je rukopis za treće izdanje i prevazišao ono što mu je bila dužnost, dajući bezbroj korisnih predloga, čak pronalazeći štamparske greške koje su postojale i u prethodnim izdanjima! Korisne predloge za određene odeljke su dali Benjamin Darwin, Mark Finkov i Igor Savin. Za sve one koje sam zaboravio da pomenem, hvala vam svima!

Bil Lewis i Mike Slinn dali su korisne komentare na više skica za prvo izdanje. Ron Hitchens i Marc Loy su pažljivo pročitali ceo konačni nacrt prvog izdanja. Zahvalan sam Miku Loukidesu za ohrabrivanje i podršku tokom celog procesa. Urednica Sue Miller pomogla je usmeravanje rukopisa tokom donekle energičnih završnih faza produkcije. Sarah Slocombe pročitala je poglavlje o XML-u u celosti i dala više sjajnih predloga; nažalost, vreme mi nije dozvolilo da ih sve uključim u prvo izdanje.

Jonathan Knudsen, Andy Oram, i David Flanagan komentarisali su konture knjige dok je još bila jedva nešto više od spiska poglavlja i recepata, i bili su u stanju da sagledaju kakva bi knjiga mogla da nastane iz toga i predlagali kako da bude još bolja.

Svaki od ovih ljudi je poboljšao knjigu na mnogo načina, pogotovo predlaganjem dodatnih recepata ili ispravljanjem postojećih. Hvala im svima! Greške koje su ostale su moje vlastite.

Čitaoci

Zahvaljujem se svim čitaocima koji su pronašli greške i predložili poboljšanja. Svako novo izdanje je bolje zahvaljujući naporima ljudi kao što ste vi koji poklonite vreme i trud da prijavite ono što treba da se prijavi!

Posebno treba pomenuti jednog od nemačkih prevodilaca knjige,[3] Gisberta Selkea, koji je pročitao prvo izdanje od korica do korica tokom prevođenja i ponegde razjasnio moj engleski tekst. Gisbert je sve to ponovio i za drugo izdanje i doprineo većem broju popravki koda, koje su ovu knjigu učinile daleko boljom nego što bi inače bila. Iako mu to nije bila dužnost, Gisbert je čak dodao jedan recept (Recept 18.5) i ispravio neke od ostalih recepata u istom poglavlju. Hvala, Gisberte!

Drugo izdanje je takođe imalo koristi od komentara Jima Burgessa, koji je pročitao velike delove knjige. Komentare o pojedinim poglavljima poslali su Jonathan Fuerth, pokojni Kim Fowler, Marc Loy i Mike McCloskey. Moja supruga, Betty, i moja deca koja su tada bila tinejdžeri su takođe lektorisali svako po nekoliko poglavlja.

Sledeći ljudi su doprineli značajnim izveštajima o greškama ili predložili poboljšanja: 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 i Patrick Wohlwend.

ltd.

Moja draga supruga, Betty Cerar, još uvek više zna o kafeinskom napitku koji pijem dok programiram nego o programskom jeziku koji koristim, ali njena strast prema jasnom izražavanju i pravilnoj gramatici je veoma koristila mom pisanju tokom našeg zajedničkog života.

Nijedna knjiga o Javi ne bi bila potpuna bez napomene zahvalnosti Jamesu Goslingu zato što je izmislio Javu (on je takođe izmislio i prvi Unix Emacs, radni list *sc* i sistem prozora NeWS). Hvala takođe njegovim poslodavcima Sun Microsystems (pre nego što ih je preuzeo Oracle) što su objavili ne samo jezik Java već i neverovatan niz Java alatki i API biblioteka besplatno dostupnih na internetu.

Willi Powell iz Apple Canada obezbedio je macOS pristup za prvo izdanje, još u ranim danima macOS-a.

Svakom od vas, moje iskreno hvala.

[3] Ranija izdanja jesu ili su bila dostupna na engleskom, nemačkom, francuskom, poljskom, ruskom, korejskom, tradicionalnom i pojednostavljenom kineskom. Zahvaljujem se svim prevodiocima na naporima da učine knjigu dostupnom široj publici.

Softver za proizvodnju knjiga

Koristio sam raznovrsne alate i operativne sisteme za pripremu, prevođenje i testiranje ove knjige. Programeri sistema OpenBSD, „proaktivno bezbednog sistema nalik na Unix", zaslužuju zahvalnost što su napravili stabilan i bezbedan klon Unixa koji je takođe bliskiji tradicionalnom Unixu od drugih besplatnih sistema. Koristio sam editor *vi* (*vi* na OpenBSD, a *vim* na Windowsu) dok sam unosio originalni rukopis u XML, a Adobe FrameMaker (jednu divnu GUI alatku za dokumentaciju koju je Adobe kupio i naknadno uništio) za formatiranje dokumenata. Ne znam da li ću ikad moći da oprostim Adobe-u što je uništio ono što je verovatno bio najbolji sistem dokumentacije na svetu i što su doveli do toga da internet postane toliko opasno mesto zadržavajući živim taj Flash zaražen greškama dugo nakon što mu je istekao vek upotrebljivosti. Samo znam da neću nikad koristiti sistem dokumentacije iz Adobe-a za bilo šta.

Zbog ovog, *Android Cookbook* koji je iz mnogo izvora i koji sam ja uređivao, nije pripremljen pomoću FrameMakera, nego pomoću XML DocBook-a (generisanog od wiki markup na veb sajtu oslonjenog na Javu koji sam napisao u tu svrhu) i jednog broja prilagođenih alata koje je obezbedila O'Reilly-jeva grupa za alate.

Treće i četvrto izdanje ovog *Java Cookbook-a* bili su formatirani u Asciidoctoru a oživeli na izdavačkom alatu O'Reilly-jevog Atlasa.