



## **PREDGOVOR**

Kao i bilo koji ljudski jezik, C++ omogućuje izražavanje ideja. Ovo sredstvo izražavanja zamišljeno je tako da veće i složenije probleme opisuje znatno jednostavnije i fleksibilnije od ostalih jezika.

C++ ne možete posmatrati samo kao skup svojstava, jer neka svojstva nemaju smisla sama za sebe. Delove možete kombinovati isključivo ako razmišljate o *projektu*, a ne samo o pisanju programa. Da biste na ovaj način razumeli C++, morate razumeti probleme C-a i programiranja uopšte. Ova knjiga razmatra programerske probleme, objašnjava zašto predstavljaju probleme i kako se na jeziku C++ pristupa njihovom rešavanju. Zato će skup mogućnosti koje objašnjavam u svakom poglavlju biti zasnovan na mom načinu posmatranja konkretnog tipa problema. Nadam se da ću vas na ovaj način, postepeno, dovesti od razumevanja C-a do tačke u kojoj principi C++-a postaju vaš prirodni način razmišljanja.

U potpunosti ću se držati stava da čitaoci treba da izgrade misaoni model koji omogućava shvatanje ovog jezika sve do nivoa mašine; ako naidete na težak problem, bićete u stanju da ga unesete u svoj model i izvedete odgovor. Pokušaću da vam prenesem saznanja koja su reorganizovala moj um tako da sam počeo da „misliti na jeziku C++“.

## Šta je novo u drugom izdanju

Ova knjiga je temeljno prerađeno prvo izdanje i odražava sve promene koje su uvedene u C++ po dovršavanju standarda i sve ono što sam ja naučio posle pisanja prvog izdanja. Celokupan tekst prvog izdanja je ispitan i prerađen. Ponekad su uklonjeni stari primeri, često su izmenjeni postojeći primeri i dodati novi, a dodati su mnogi novi zadaci. Značajno preuređenje i promena redosleda izlaganja odražavaju dostupnost boljih alatki i moje bolje razumevanje načina na koji ljudi uče C++. Dodato je novo poglavlje, koje predstavlja brzi uvod u koncepte jezika C i osnovne mogućnosti jezika C++, za čitaoce bez predznanja C-a neophodnog da bi razumeli ostatak knjige. Prateći CD sadrži seminar, još postupniji uvod u programiranje na C-u, što je neophodno za shvatanje C++-a (ili Jave). Pripremio ga je Chuck Allison za moju kompaniju (Mind View, Inc.), a naslov je *Thinking in C: Foundations for Java and C++ (Misliti na C-u: osnove Jave i C++-a)*. Seminar uvodi korisnika u aspekte C-a neophodne za prelazak na C++ ili Javu, bez ulaženja u teške detalje kojima se C programeri svakodnevno bave, ali nisu bitni za programere na jezicima C++ i Java.

Kratak odgovor na pitanje: „U čemu se razlikuje drugo izdanje?“ glasi „ono što nije sasvim novo, prerađeno je, ponekad toliko da ne prepoznajete prvobitne primere i materijal“.

## Šta se nalazi u drugom tomu ove knjige

Upotpunjeni standard C++-a dodao je i više novih značajnih biblioteka, koje obuhvataju klasu **string**, kontejnere i algoritme standardne C++ biblioteke, kao i šablone. Napredne teme su prebačene u drugi tom ove knjige, uključujući i važna pitanja kao što su višestuko nasleđivanje, obrada izuzetaka, obrasci projektovanja i teme o izgradnji i otkrivanju grešaka u stabilnim sistemima.

## Kako doći do drugog toma

Elektronsku verziju drugog toma knjige *Misliti na jeziku C++*, na engleskom jeziku, naći ćete na mojoj Web lokaciji *www.BruceEckel.com*. Na ovoj adresi naći ćete i podatke o očekivanom datumu objavljivanja konačne verzije toma 2.

Ova Web lokacija sadrži i izvorni kôd primera iz oba toma, kao i izmene i informacije o drugim seminarima na CD-ovima kompanije MindView.

## Preduslovi

U prvom izdanju ove knjige, pretpostavljao sam da ste od nekog drugog naučili C i da bar lako čitate kôd. Usredsredio sam se na pojednostavljenje onoga što sam smatrao teškim: jezika C++. U ovo izdanje sam dodao poglavlje koje brzo uvodi u C, uz seminar na CD-u, ali i dalje pretpostavljam da već imate programersko iskustvo. Osim toga, kao što intuitivno učite nove reči stranih jezika kada ih vidite u određenom kontekstu u romanu, moguće je puno naučiti o C-u iz primera u knjizi.

## Učenje C++-a

Svoj put u C++ prokrčio sam sa iste pozicije u kojoj očekujem da se nalaze mnogi čitaoci ove knjige: programer s neposrednim, praktičnim stavom prema programiranju. Što je još teže, pre toga sam se bavio programiranjem hardvera, u kome se C često smatrao jezikom visokog nivoa i neefikasnim preterivanjem u manipulaciji bitovima. Kasnije sam otkrio da čak nisam bio ni naročito dobar C programer. Krio sam da ne poznajem strukture, funkcije **malloc()** i **free()**, **setjmp()** i **longjmp()** i druge „napredne“ koncepte, bežeći posramljeno kada se u razgovoru dođe do ovih tema, umesto da tragam za novim saznanjima.

Kada sam se uhvatio u koštac s jezikom C++, jedina pristojna knjiga bila je „vodič za stručnjake“<sup>1</sup> – barem ju je tako naslovio njen autor Bjarne Stroustrup – tako da sam morao sam sebi da razjašnjavam osnovne koncepte. To je dovelo do moje prve knjige o C++-u,<sup>2</sup> koja je u suštini bila ispovest o sopstvenom iskustvu. Bila je organizovana kao vodič za čitaoca, namenjena istovremenom uvođenju u C i C++. Čitaoci su oduševila oba izdanja ove knjige.<sup>3</sup>

Otrprike u vreme kada je objavljena knjiga *Using C++*, počeo sam da predajem ovaj jezik na seminarima i prezentacijama. Podučavanje C++-a (kasnije i Java) postalo je moja profesija; širom sveta sam od 1989. gledao publiku koja klima glavom, bezizražajna lica i s nerazumevanjem. Kada sam počeo da obučavam manje grupe unutar preduzeća, na vežbama sam došao do jednog otkrića. Čak i one koji su se smeškali i klimali glavama zbunjivala su mnoga važna pitanja. Tokom višegodišnjeg organizovanja i predsedavanja odsecima za C++ i Javu na Konferenciji za razvoj softvera, otkrio sam da smo i drugi govornici i ja težili da pružimo određenoj publici što više informacija za što kraće vreme. Usled razlika u nivou znanja publike, odnosno načina izlaganja materijala, do kraja bih izgubio

<sup>1</sup> Bjarne Stroustrup: *The C++ Programming Language*, Addison-Wesley, 1986. (prvo izdanje)

<sup>2</sup> *Using C++*, Osborne/McGraw-Hill, 1989.

<sup>3</sup> *Using C++ i C++ Inside & Out*, Osborne/McGraw-Hill, 1993.

deo publike. Možda je to previše zahtevno, ali kako imam otpor prema tradicionalnim predavanjima (verujem da je, u većini slučajeva, ovaj otpor posledica dosade), hteo sam da svima održim pažnju.

Neko vreme sam organizovao više različitih izlaganja u prilično kratkom periodu. Tako sam na kraju učio kroz eksperimente i ponavljanje (ista tehnika je sasvim primenljiva i na projektovanje programa). Na kraju sam razvio stil podučavanja koristeći sve što sam naučio kroz svoje iskustvo predavača. Problem učenja se savladava u posebnim, lako svarljivim koracima, a na seminaru sa vežbama (savršen način učenja) postoje zadaci koji prate svako izlaganje. Informacije o mojim javnim seminarima naći ćete na adresi [www.BruceEckel.com](http://www.BruceEckel.com), a tu možete nabaviti i seminare na CD-ovima.

Prvo izdanje ove knjige razvijano je dve godine, a obuhvaćeni materijal je testiran u praksi, u više oblika, na mnogim različitim seminarima. Povratne informacije koje sam dobijao sa svakog seminarara pomagale su mi da menjam i reorganizujem ovaj materijal, sve dok nisam stekao utisak da je postao pogodno sredstvo za podučavanje. Međutim, ovo nisu samo beleške sa seminarara; pokušao sam da na ove stranice smestim što više informacija i da ih organizujem tako da vas vode do sledeće teme. Knjiga je, pre svega, zamišljena tako da posluži pojedincu koji se bori s novim programskim jezikom.

## Ciljevi

Ciljevi ove knjige su da:

1. Izloži materijal u pojedinačnim jednostavnim koracima, tako da čitalac lako može da savlada svaku ideju pre nego što pređe na sledeću.
2. Predstavi što jednostavnije i kraće primere. Ovo me često sprečava da se bavim problemima iz „stvarnog sveta“, ali sam otkrio da početnici obično više vole da razumeju svaki detalj jednog primera nego da budu impresionirani opsegom problema koji on rešava. Osim toga, postoji ozbiljno ograničenje količine koda koji se može usvojiti u učionici. Zato ponekad dobijam kritike za korišćenje „ primera za igru“, ali sam spreman da ih prihvatim jer smatram da su pedagoški opravdani.
3. Pažljivo rasporedi izlaganje mogućnosti jezika, tako da ne vidite nešto čemu niste ranije bili izloženi. Naravno da to nije uvek moguće; u nekim situacijama biće dat sažet uvodni opis.
4. Pruži čitaocima podatke o jeziku koje smatram važnim, a ne sve što ja znam. Verujem da postoji „hijerarhija značaja informacija“, a postoje i činjenice koje 95 procenata programera nikada neće morati da zna i koje bi ih samo zbunile i navele da pomisle da je C++ suviše složen. Evo primera iz C-a: ako naučite napamet tabelu prioriteta operatora (lično je nisam nikada naučio), možete vešto pisati izraze. Ali, ako *vi* morate da razmišljate o tome šta ti izrazi rade, onaj ko čita/održava kôd biće zbunjen. Zato zaboravite prioritete i koristite zagrade ako vam stvari nisu sasvim jasne. Isti stav će biti zauzet i u pogledu nekih informacija o C++-u, za koje mislim da su važnije projektantima prevodilaca nego programerima.

5. Usredsredi svaki odeljak tako da vreme potrebno za lekciju i razmak između vežbanja bude prihvatljiv. Time se održava aktivnost i pažnja publike tokom seminara sa vežbama, a čitaocu se pruža bolji osećaj šta je savladao.
6. Obezbedi čitaocima dobru osnovu, tako da dovoljno dobro shvate važna pitanja, kako bi prešli na složenije kurseve i knjige (konkretno, drugi tom ove knjige).

Pokušao sam da izbegnem verzije C++-a konkretnih proizvođača, pošto mislim da pojedinosti određene realizacije nisu toliko važne za učenje koliko sam jezik. Dokumentacija većine prevodilaca je sasvim odgovarajuća za uvod u njihove specifičnosti.

## Poglavlja

C++ je jezik u kome su nove i drugačije mogućnosti izgrađene povrh postojeće sintakse. (S obzirom na ovo, naziva se *hibridnim* objektno orijentisanim programskim jezikom.) Posmatrajući kako ljudi uče ovaj jezik, stekao sam osećaj za način na koji programeri prolaze kroz nivoe mogućnosti jezika C++. Zbog toga što to deluje kao prirodno napredovanje uvežbanog proceduralnog načina mišljenja, odlučio sam da sledim istu putanju i podstičem proces postavljanja i odgovaranja na pitanja koja su se pojavljivala dok sam učio jezik ili su stizala iz publike dok sam taj jezik predavao.

Ovaj tečaj je organizovan sa jednom stvari na umu: da se racionalizuje proces učenja C++-a. Povratne informacije od publike pomogle su mi da razumem koji su delovi bili teški i zahtevali dodatno objašnjenje. Shvatio sam sledeće: ako preambiciozno uključite veliki broj novih mogućnosti, morate ih sve i objasniti, da studenti ne bi bili zbunjeni. Rezultat je da sam uložio veliki napor da uvodim što manje novih osobina odjednom, a idealno je da postoji jedan osnovni koncept po poglavlju.

Cilj svakog poglavlja je da obradi jednu temu ili mali broj povezanih tema, tako da ne zavise od bilo kakvih dodatnih mogućnosti. Na taj način svaki deo možete usvojiti pre nego što krenete dalje. Da bih to postigao, pominjem neka svojstva jezika C duže nego što bih želeo. Korist od ovog pristupa je što vas neće zbuniti upotreba svih mogućnosti C++-a pre nego što budu objašnjene, tako da će uvođenje u jezik biti postupno i odražavaće način na koji biste samostalno usvajali ove mogućnosti.

Sledi sažet opis poglavlja obuhvaćenih knjigom:

**Poglavlje 1: Uvod u objekte.** Kada su projekti postali previše veliki a njihovo održavanje suviše komplikovano, nastala je „softverska kriza“, a programeri su govorili: „Ne možemo završiti projekte, a i preskupi su!“ To je izazvalo niz odgovora koji se razmatraju u ovom poglavlju, uporedo sa idejama o objektno orijentisanom programiranju (OOP) i kako bi ono moglo rešiti softversku krizu. Ovo poglavlje vas vodi kroz osnovne koncepte i mogućnosti OOP-a i uvodi postupke analize i projektovanja. Saznaćete i prednosti i značaj usvajanja ovog jezika i preporuke za prelazak u svet C++-a.

**Poglavlje 2: Pravljenje i upotreba objekata.** Ovo poglavlje objašnjava proces izgradnje programa pomoću prevodilaca i biblioteka. Predstavlja se prvi C++ program u knjizi i pokazuje se kako se pišu i prevode programi. Zatim se uvode neke od osnovnih biblioteka objekata, dostupnih u C++-u. Kad stignete do kraja ovog poglavlja, biće vam prilično jasno šta znači koristiti gotove biblioteke objekata prilikom pisanja programa na jeziku C++.

**Poglavlje 3: C u C++-u.** Ovo poglavlje daje sažeti pregled osobina C-a koje se koriste u C++-u, kao i više osnovnih mogućnosti dostupnih samo u C++-u. Predstavljam i popularan uslužni program `make` koji se koristi za pravljenje izvršnih verzija svih primera navedenih u knjizi (izvorni kôd primera iz ove knjige, dostupan na adresi [www.BruceEckel.com](http://www.BruceEckel.com), sadrži datoteke `makefile` za svako poglavlje). Poglavlje 3 pretpostavlja da imate solidne osnove iz nekog proceduralnog jezika, kao što su Pascal, C, čak i neke verzije Basica (ako ste u tom jeziku napisali mnogo koda, naročito funkcija). Ako vam je ovo poglavlje preteško, prvo bi trebalo da prođete kroz seminar *Thinking in C*, koji se nalazi na priloženom CD-u (dostupan je i na [www.BruceEckel.com](http://www.BruceEckel.com)).

**Poglavlje 4: Apstrakcija podataka.** Većina svojstava C++-a odnosi se na pravljenje novih tipova podataka. To omogućuje bolju organizaciju koda i postavlja osnove naprednijih mogućnosti OOP-a. Videćete kako se ova ideja primenjuje jednostavnim smeštanjem funkcija u strukture, kako možete praviti nove tipove i kakav kôd se dobija. Upoznaćete i najbolji način organizovanja koda u datoteke zaglavlja i datoteke realizacije.

**Poglavlje 5: Skrivanje realizacije.** Možete odlučiti da neki podaci i funkcije u strukturi budu nedostupni korisniku novog tipa tako što ćete ih proglasiti privatnim. To znači da možete odvojiti realizaciju od interfejsa koji vidi programer klijent i tako omogućiti jednostavnu izmenu realizacije bez narušavanja klijentovog koda. Uvodi se i rezervisana reč **class** kao elegantniji način opisivanja novog tipa podatka, a objašnjava se i značenje reči „objekat“ (to je novi tip promenljive).

**Poglavlje 6: Inicijalizacija i čišćenje memorije.** Jedna od najčešćih grešaka u programima na C-u potiče od neinicijalizovanih promenljivih. U C++-u vam *konstruktor* omogućava da garantujete da će promenljive novog tipa podatka („objekti vaše klase“) uvek biti ispravno inicijalizovane. Ako objekti zahtevaju i brisanje, možete garantovati da će se ono uvek obaviti pomoću *destruktor*a.

**Poglavlje 7: Preklapanje funkcija i podrazumevani argumenti.** Namena C++-a je da vam pomogne da izgradite velike, složene projekte. Za vreme rada možete koristiti više biblioteka sa istoimenim funkcijama, a možete odlučiti i da, u okviru iste biblioteke, koristite isto ime sa različitim značenjima. C++ to olakšava *preklapanjem funkcija*, što omogućava da više funkcija ima isto ime, sve dok se razlikuju liste njihovih argumenata. Podrazumevani argumenti omogućavaju da na različite načine pozovete istu funkciju, automatski obezbeđujući podrazumevane vrednosti za neke od argumenata.

**Poglavlje 8: Konstante.** Ovo poglavlje obrađuje rezervisane reči **const** i **volatile**, koje imaju dodatno značenje u C++-u, naročito unutar klase. Saznaćete šta znači primeniti **const** na definiciju pokazivača. U ovom poglavlju se pokazuje i

kako se značenje reči **const** razlikuje ako se koristi u klasama i izvan njih i kako u vreme prevodenja napraviti konstante unutar klasa.

**Poglavlje 9: Umetnute funkcije.** Pretprocesorski makroi uklanjaju suvišne pozive funkcija, ali pretprocesor zaobilazi i dragocenu proveru tipova jezika C++. Umetnuta funkcija pruža sve prednosti pretprocesorskog makroa i sve prednosti pravog poziva funkcije. Ovo poglavlje detaljno istražuje realizaciju i upotrebu umetnutih funkcija.

**Poglavlje 10: Upravljanje imenima.** Definisavanje imena je osnovna aktivnost u programiranju, a kako projekat raste, broj imena može postati prevelik. C++ vam omogućava priličnu kontrolu nad imenima, pa možete upravljati definisanjem, vidljivošću, smeštanjem u memoriju i povezivanjem. Ovo poglavlje pokazuje dve tehnike upravljanja imenima. Prvo, rezervisana reč **static** se koristi za kontrolu vidljivosti i povezivanja, a ispituje se i njeno specijalno značenje u klasama. Mnogo korisnija tehnika za upravljanje imenima u globalnom opsegu jeste korišćenje rezervisane reči **namespace**, koja omogućava da globalni imenski prostor podelite u posebne oblasti.

**Poglavlje 11: Reference i konstruktor za kopiranje.** Pokazivači jezika C++ su slični pokazivačima jezika C, uz dodatnu prednost strože provere tipova u C++-u. C++ obezbeđuje i dodatni način rukovanja adresama: iz Algola i Pascala preuzima *referencu* koja prepušta prevodiocu rad s adresama dok vi koristite uobičajeni zapis. Susrećete se i s konstruktorom za kopiranje, koji određuje način prenošenja objekata po vrednosti u funkcije i iz funkcija. Na kraju se razjašnjava i C++-ov pokazivač na član.

**Poglavlje 12: Preklapanje operatora.** Ova mogućnost se ponekad naziva „zaslađivač sintakse“; omogućava vam da pojednostavite sintaksu upotrebe tipa dozvoljavajući pozive operatora kao i funkcija. U ovom poglavlju ćete saznati da je preklapanje operatora samo tip preklapanja funkcija i naučićete kako sami da preklopite operatore. Naučićete kako da navedete argumente, kako da odredite tip povratne vrednosti i kako da donesete odluku da li operator treba da bude član ili prijatelj klase.

**Poglavlje 13: Dinamičko stvaranje objekata.** Kolikim brojem aviona treba da upravlja sistem za kontrolu leta? Koliko oblika će sadržati jedan CAD sistem? U opštim programerskim problemima ne možete znati količinu, vek trajanja i tip objekata potrebnih programu za vreme izvršavanja. U ovom poglavlju ćete saznati kako rezervisane reči **new** i **delete** elegantno rešavaju ovaj problem, bezbedno praveći objekte u dinamičkom memorijskom prostoru. Videćete i kako se operatori **new** i **delete** mogu na različite načine preklopiti, tako da možete kontrolisati dodeljivanje i oslobađanje memorijskog prostora.

**Poglavlje 14: Nasleđivanje i slaganje.** Apstrakcije podataka omogućavaju da definišete nove tipove od početka, ali pomoću omotavanja i nasleđivanja možete napraviti nove tipove od postojećih. Slaganjem možete napraviti nov tip čiji su delovi postojeći tipovi, a nasleđivanjem pravite specijalnu verziju postojećeg tipa. U ovom poglavlju ćete naučiti sintaksu nasleđivanja i slaganja, način redefinisavanja funkcija i značaj konstruktora i destruktora.

**Poglavlje 15: Polimorfizam i virtuelne funkcije.** Možda bi vam bilo potrebno čitavih devet meseci za samostalno otkrivanje i razumevanje ovih osnovnih pojmova OOP-a. Kroz kratke i jednostavne primere, videćete kako da uz nasleđivanje napravite porodicu tipova i kako da radite s objektima porodice preko njihove zajedničke osnovne klase. Rezervisana reč **virtual** omogućava zajednički rad sa svim srodnim tipovima objekata, tako da se veliki deo koda ne oslanja na informacije o određenom tipu. To doprinosi proširivosti programa, pa je pisanje i održavanje koda jednostavnije i jeftinije.

**Poglavlje 16: Uvod u šablone.** Nasleđivanje i slaganje omogućavaju da ponovo koristite objektni kôd, ali ne zadovoljavaju sve potrebe za ponovnom upotrebom koda. Šabloni omogućavaju ponovnu upotrebu *izvornog* koda dajući prevodiocu sredstvo za zamenu imena tipova u telu klase ili funkcije. To je primenjeno na biblioteku klasa *kontejnera*, koje su važne alatke za brz razvoj moćnih objektno orijentisanih programa (standardni C++ obuhvata značajnu biblioteku klasa *kontejnera*). Poglavlje vam daje kompletnu osnovu ove bitne teme.

Dodatne teme (i napredniji sadržaji) dostupni su u drugom tomu ove knjige, čija se radna verzija na engleskom jeziku može preuzeti s Web lokacije [www.BruceEckel.com](http://www.BruceEckel.com).

## Zadaci

Otkrio sam da su zadaci posebno korisni da bi se tokom seminara upotpunilo znanje polaznika, tako da ćete ih pronaći na kraju svakog poglavlja. Broj zadataka je mnogo veći nego u prvom izdanju.

Mnogi zadaci su sasvim jednostavni, tako da se mogu uraditi relativno brzo na času, pod nadzorom predavača, čime se obezbeđuje da svi polaznici usvoje gradivo. Neki zadaci su teži, kako bi se održala pažnja naprednijih polaznika. Većina zadataka je sastavljena tako da se može brzo rešiti, namenjena je samo ispitivanju i upotpunjavanju znanja, a ne predstavlja veliki izazov (verovatno ćete takve pronaći sami ili, verovatnije, oni će pronaći vas).

## Rešenja zadataka

Rešenja izabranih zadataka se nalaze u elektronskom dokumentu *The Thinking in C++ Annotated Solution Guide* koji se, uz malu nadoknadu, može preuzeti s lokacije [www.BruceEckel.com](http://www.BruceEckel.com).

## Izvorni kôd

Izvorni kôd primera iz knjige zaštićen je autorskim pravom i besplatan, a distribuira se preko Web lokacije [www.BruceEckel.com](http://www.BruceEckel.com). Zaštita autorskog prava vas sprečava da reprodukujete kôd u štampanom obliku bez dozvole, ali imate pravo da ga koristite u mnogim drugim situacijama.



Kôd je spakovan u arhivu i može se raspakovati na svakom sistemu koji ima uslužni program zip (većina ima, a ako nije već instaliran, na Internetu možete pronaći verziju pogodnu za svoj sistem). U direktorijumu gde ste raspakovali kôd, pronaći ćete sledeću napomenu o autorskom pravu (na engleskom):

```
///  
:Copyright.txt  
Copyright (c) 2000, Bruce Eckel  
Izvorni kôd iz knjige „Misliti na jeziku C++“  
Sva prava zadržana, OSIM sledećih: Ovu datoteku možete slobodno kori-  
stiti u svom radu (ličnom ili komercijalnom), uključujući izmene i  
distribuciju isključivo u izvršnom obliku. Odobrava se korišćenje ove  
datoteke u obuci, uključujući i prezentacije, ali se kao izvor mora  
navesti knjiga „Misliti na jeziku C++“. Osim radi obuke, ovaj kôd ne  
smete kopirati niti distribuirati; može se preuzeti jedino sa adrese  
http://www.BruceEckel.com (i zvanične kopije servera), gde je besplatno  
dostupan. Ovu oznaku i napomenu o autorskom pravu ne smete brisati. Ne  
smete distribuirati izmenjene verzije izvornog koda iz ovog paketa. Ovu  
datoteku ne smete koristiti u štampanom obliku bez izričitog odobrenja  
autora. Bruce Eckel ne tvrdi da je ovaj softver pogodan za bilo koju  
svrhu. Priložen je u ovom obliku, bez izričite ili podrazumevane gara-  
ncije bilo koje vrste, uključujući bilo kakvu podrazumevanu garanciju  
tržišne vrednosti, pogodnosti za određenu svrhu ili kršenje nečijih  
prava. Preuzimate kompletan rizik za korišćenje ovog softvera. Ni Bruce  
Eckel ni izdavač neće biti odgovorni za bilo kakva oštećenja naneta vama  
ili trećem licu koja su rezultat upotrebe ili distribucije ovog soft-  
vera. Ni u jednoj situaciji Bruce Eckel ni izdavač neće biti odgovorni  
za bilo kakav gubitak prihoda, profita, podataka ili za neposrednu,  
posrednu, posebnu, posledičnu, slučajnu ili krivičnu štetu, nastalu pod  
bilo kojim okolnostima i bez obzira na teoriju o odgovornosti koja proi-  
zlazi iz upotrebe ili nemogućnosti upotrebe softvera, čak i ako su Bruce  
Eckel i izdavač bili upozoreni na mogućnost takvih šteta. Ako se ispolje  
greške ovog softvera, vi snosite troškove svih neophodnih usluga i  
popravki. Ako mislite da ste otkrili grešku, molim da predložite ispra-  
vku preko obrasca koji ćete pronaći na adresi www.BruceEckel.com. (Molim  
da koristite isti obrazac i za otkrivene greške u tekstu ove knjige.)  
///  
:~
```

Kôd možete koristiti u svojim projektima i pri obuci sve dok se poštuje napomena o autorskom pravu.

## Standardi jezika

U ovoj knjizi ću navesti samo 'C' kada podrazumevam da verzija jezika odgovara ISO C standardu. Praviću razliku samo ako je neophodno razgraničiti standardni C i starije verzije.

U vreme pisanja ove knjige, Komitet za standardizaciju C++-a završio je rad na ovom jeziku. Koristiću termin *standardni C++* kada ukazujem na standardizovani jezik. Kada je napisano samo C++, znajte da podrazumevam „standardni C++“.

Postoji izvesna zabuna oko stvarnog naziva Komiteta za standardizaciju C++-a i samog naziva standarda. Predsedavajući komiteta, Steve Clamage, razjasnio je ovo:

*Postoje dva komiteta za standardizaciju jezika C++: komitet NCITS (prethodno X3) J16 i komitet ISO JTC1/SC22/WG14. ANSI ovlašćuje NCITS da formira tehničke komitete za razvoj američkih nacionalnih standarda.*

*J16 je 1989. ovlašćen da definiše američki standard za C++. Oko 1991. je WG14 ovlašćen da napravi međunarodni standard. Projekat J16 je pretvoren u „Tip I“ (internacionalni) i podređen radu na ISO standardizaciji.*

*Ova dva komiteta se sastaju u isto vreme na istom mestu, a J16 glasanjem obrazuje američki deo WG14. WG14 prenosi tehnički deo posla komitetu J16. WG14 glasanjem odlučuje o tehničkom radu komiteta J16.*

*Prvobitno je C++ standard formiran kao ISO standard. ANSI je kasnije izglasao (po preporuci komiteta J16) usvajanje ISO C++ standarda kao američkog standarda za jezik C++.*

Zato je ISO ispravan način opisivanja standarda C++-a.

## Podrška jeziku

Vaš prevodilac možda ne podržava sve mogućnosti razmotrene u ovoj knjizi, naročito ako nemate najnoviju verziju. Realizacija prevodioca za jezik kao što je C++ predstavlja herkulovski poduhvat i možete očekivati da će se osobine jezika pojavljivati po grupama, a ne sve odjednom. Ako isprobate neki primer iz knjige i prevodilac vam prijavi mnoštvo grešaka, to ne mora ukazivati na grešku u kodu ili grešku u prevodiocu – možda prevodilac još uvek ne podržava ovaj primer.

## Prateći kompakt disk

Osnovni sadržaj pratećeg CD-a je „seminar na CD-u“ pod nazivom *Thinking in C: Foundations for Java and C++ (Misliti na C-u: osnove Jave i C++-a)* autora Chucka Allisona. CD je objavila kompanija Mind View, a seminar se može preuzeti i sa adrese [www.BruceEckel.com](http://www.BruceEckel.com). Sadrži više časova audio lekcija i slajdova i može se pregledati na većini računara ako imate čitač kompakt diska i audio karticu.

Cilj seminara je da vas pažljivo provede kroz osnove jezika C. Usredsređuje se na znanje koje vam je neophodno da biste bili u stanju da predete na jezike C++ ili Java, umesto da nastoji da od vas napravi stručnjaka za sve pojedinosti jezika C. (Jedan od razloga za korišćenje jezika višeg nivoa, kao što su C++ ili Java, jeste upravo mogućnost izbegavanja ovih pojedinosti.) Sadrži i zadatke i uputstva za njihovo rešavanje. Imajte na umu da, pošto poglavlje 3 ove knjige po obimu prevazilazi seminar na kompakt disku, CD nije zamena za to poglavlje, nego bi ga trebalo koristiti kao pripremu za knjigu.

Pošto je za gledanje sadržaja sa kompakt diska potreban Web čitač, trebalo bi da, pre upotrebe CD-a, instalirate čitač.

## Kompakt diskovi, seminari i savetovanja

Predviđeno je da seminari na kompakt diskovima pokrivaju tom 1 i tom 2 ove knjige. Oni obuhvataju više sati zvučnih zapisa mojih predavanja, uz slajdove koji sadrže gradivo odabrano iz svakog poglavlja knjige. Ove kompakt diskove možete poručiti na adresi [www.BruceEckel.com](http://www.BruceEckel.com), gde ćete pronaći više informacija i primera lekcija.

Kompanija MindView organizuje javne seminare sa praktičnim vežbama, zasnovane na materijalu izloženom u ovoj knjizi i na naprednijim temama. Lekcije su sačinjene od izabranog materijala iz svakog poglavlja, posle čega sledi vežbanje uz nadzor, tako da se svakom pojedinačnom polazniku posvećuje pažnja. Mi obezbeđujemo i obuku, savetovanje, vođenje i prolazak kroz projekat i kôd na licu mesta. Informacije i prijavni obrasci za predstojeće seminare, kao i drugi podaci za kontakt, mogu se naći na adresi [www.BruceEckel.com](http://www.BruceEckel.com).

Ja pružam i savetodavne usluge u projektovanju, proceni projekta i pregledanju koda. Kada sam počeo da pišem o računarima, osnovni motiv mi je bio da se više bavim savetovanjem, pošto mislim da je to izazovno, poučno i jedno od mojih najprijetnijih profesionalnih iskustava. Zato ću dati sve od sebe da vas uklopim u svoj raspored ili da vam obezbedim nekog od svojih saradnika (to su ljudi koje dobro poznajem i kojima verujem, a često učestvuju i u organizovanju i izvođenju seminara).

## Greške

Bez obzira na to koliko veštine pisac uloži da bi otkrio greške, neke se uvek provuku i pravo sa stranice zaskoče novog čitaoca. Ako otkrijete bilo šta što smatrate greškom, molim da popunite i pošaljete obrazac za ispravke, koji ćete pronaći na adresi [www.BruceEckel.com](http://www.BruceEckel.com). Poštujem vašu pomoć.

## O koricama

Na koricama prvog izdanja ove knjige bio je moj lik. Želeo sam da korice drugog izdanja budu više u umetničkom duhu, poput korica knjige *Misliti na Javi*. Iz nekog razloga, C++ mi je nametnuo ideju da se upotrebi stil Art Deco sa jednostavnim linijama i „hromiranim“ površinama. Ja sam hteo da korice liče na postere brodova i aviona sa dugim telima u pokretu.

Moj prijatelj Daniel Will-Harris ([www.Will-Harris.com](http://www.Will-Harris.com)), koga sam upoznao u horu osnovne škole, postao je dizajner i pisac svetskog nivoa. On mi je uradio skoro sav dizajn, uključujući korice za prvo izdanje ove knjige. Dok je osmišljavao korice, nezadovoljan napretkom, Danijel se neprekidno pitao: „Kako ovo povezuje ljude i računare?“ Bili smo u klopci.

Sasvim iznenada, bez ideje o konačnom ishodu, zatražio je da stavim lice na skener. Daniel je u programu za obradu slika Corel Xara vektorizovao moj lik. Po njegovim rečima, „automatska vektorizacija je način na koji računar pretvara sliku u linije i krive po svom ukusu“. Zatim se igrao s tim sve dok rezultat nije postao nalik na topografsku mapu mog lica, sliku koja bi mogla predstavljati način na koji računar vidi ljude.

Uzeo sam ovu sliku i fotokopirao je na papir za akvarele (neki fotokopir aparati u boji mogu da prime i tako debeo materijal), a zatim sam dosta eksperimentisao dodajući slici vodene boje. Izabrali smo slike koje su nam se najviše svidele, zatim ih je Daniel ponovo skenirao i od njih napravio korice, dodajući tekst i druge elemente. Ceo proces je trajao nekoliko meseci, uglavnom zbog vremena koje sam utrošio slikajući. Posebno sam uživao u tome, pošto sam učestvovao u stvaranju dela na koricama, a i zato što mi je to dalo podsticaj da crtam vodenim bojama (zaista je tačno ono što kažu o vežbanju).

## Dizajn i priprema knjige

Unutrašnjost knjige dizajnirao je Daniel Will-Harris, koji se u osnovnoj školi igrao samolepljivim slovima, čekajući pronalazak računara i stonog izdavaštva. Fotoslog sam lično napravio, tako da su i greške uređivanja teksta moje. Za pisanje knjige i pravljenje fotosloga, kao i za generisanje sadržaja i indeksa, korišćen je Microsoftov Word za Windows, verzije 8 i 9. (U Pythonu sam napravio COM komponentu, koja se poziva iz VBA makroa, kako bi mi pomogao u označavanju indeksa). Python (videti [www.Python.org](http://www.Python.org)) je korišćen za pisanje nekih alati za proveru koda i bio bi upotrebljen kao alat za izdvajanje koda da sam ga otkrio ranije.

Dijagrame sam napravio koristeći Visio – hvala kompaniji Visio Corporation zato što je napravila korisnu alatku.

Konačna verzija fotosloga napravljena je pomoću Adobe Acrobat 4 i poslata je u štampu neposredno iz te datoteke – mnogo hvala kompaniji Adobe na alatki koja dopušta slanje dokumenata spremnih za štampu elektronskom poštom, pošto to omogućava veći broj revizija u jednom danu, tako da ne moram da se oslanjam na svoj laserski štampač i noćne poštanske usluge. (Postupak sa Acrobatom isprobali smo na knjizi *Misliti na Javi*, tako da sam konačnu verziju te knjige mogao iz Južne Afrike preneti na štampu u Sjedinjene Države.)

HTML verzija je napravljena pretvaranjem dokumenta iz Worda u format RTE, a zatim je RTF2HTML (videti <http://www.sunpack.com/RTF/>) uradio najveći deo konverzije u HTML. Hvala Chrisu Hectoru zato što je napravio tako koristan i veoma pouzdan alat. Dobijene datoteke su prečišćene upotrebom programa na Pythonu, a WMF datoteke su pretvorene u format GIF pomoću programa JASC PaintShop Pro 6 i njegovog alata za paketnu konverziju (hvala kompaniji JASC, čiji je odličan proizvod rešio mnoštvo mojih problema). Perl skript za označavanje sintakse različitim bojama ljubazni je doprinos Zafira Anjuma.

## Zahvalnice

Prvo, hvala svima koji su putem Interneta poslali ispravke i predloge; vaša pomoć je bila dragocena za poboljšanje kvaliteta ove knjige i to ne bih mogao obaviti bez vas. Posebnu zahvalnost upućujem Johnu Cooku.

Ideje i podrška za ovu knjigu došli su iz mnogih izvora: od prijatelja, kao što su Chuck Allison, Andrea Provaglio, Dan Saks, Scott Meyers, Charles Petzold i Michael Wilk; začetnika ovog jezika, kao što su Bjarne Stroustrup, Andrew Koenig i Rob Murray; od članova Komiteta za standardizaciju C++-a, među kojima su Nathan Myers (njegova zapažanja su bila posebno korisna i velikodušno ih je saopštio), Bill Plauger, Reg Charney, Tom Penello, Tom Plum, Sam Druker i Uwe Steinmueller; od ljudi koji su govorili na mom odseku za C++ na Konferenciji za razvoj softvera; često i od polaznika mojih seminara, koji su postavljali pitanja koja su mi bila potrebna da bih ovaj materijal učinio jasnijim.

Posebnu zahvalnost zaslužuje moj prijatelj Gen Kiyooka, čija mi je kompanija Digigami obezbedila mrežni server.

Moj prijatelj Richard Hale Shaw i ja zajedno smo predavali C++; Richardova pronicljivost i podrška bili su veoma korisni (to važi i za Kim). Zahvalnost zaslužuju i KoAnn Vikoren, Eric Faurot, Jennifer Jessup, Tara Arrowood, Marco Pardi, Nicole Freeman, Barbara Hanscome, Regina Ridley, Alex Dunne i ostalo društvo iz MFI-ja.

Posebno zahvaljujem svim svojim učiteljima i svim svojim studentima (koji su istovremeno i moji učitelji).

Duboko poštovanje izražavam svojim omiljenim piscima. To su: John Irving, Neal Stephenson, Robertson Davies (nedostajacete nam), Tom Robbins, William Gibson, Richard Bach, Carlos Castaneda i Gene Wolfe.

Zahvalnost zaluzuje Guido van Rossum koji je izumeo Python i velikodušno ga dao svetu. Svojim doprinosom ste obogatili moj život.

Hvala ljudima iz Prentice Halla. To su: Alan Apt, Ana Terry, Scott Disanno, Toni Holm i moj redaktor elektronske kopije, Stephanie English, a u marketingu Bryan Gambrel i Jennie Burger.

Sonda Donovan je pomogla u proizvodnji kompakt diska. Daniel Will-Harris (naravno) raskošno je dizajnirao sâm disk.

Zahvalnost dugujem divnim ljudima koji su Crested Butte učinili izuzetnim mestom, a to su naročito Al Smith (tvorac sjajnog Camp4 Coffee Gardena), moji susedi Dave i Erika, Marsha iz knjižare Heg's Place, Pat i John iz Teocalli Tamale, Sam iz Bakery Caféa i Tiller koji mi je pomogao u istraživanju zvuka. Hvala i svim divnim ljudima iz Camp4 koji su čine zanimljivim moja jutra.

Na listi prijatelja koji me podržavali nalaze se, između ostalih, i: Zack Urlocker, Andrew Binstock, Neil Rubenking, Kraig Brockschmidt, Steve Sinofsky, JD Hildebrandt, Brian McElhinney, Brinkley Barr, Larry O'Brien, Bill Gates iz časopisa *Midnight Engineering*, Larry Constantine, Lucy Lockwood, Tom Keffer, Dan Putterman, Gene Wang, Dave Mayer, David Intersimone, Claire Sawyers, Italijani (Andrea Provaglio, Rossella Gioia, Laura Fallai, Marco i Lella Cantu, Colorado, Ilsa i Christina Giustozzi), Chris i Laura Strand (i Parker), porodica Almqvist, Brad Jerbic, Marilyn Cvitanic, porodica Mabry, porodica Haflinger,

porodica Pollock, Peter Vinci, porodica Robbins, porodica Moelters, Dave Stoner, Laurie Adams, porodica Cranston, Larry Fogg, Mike i Karen Sequeira, Gary Entsminger i Allison Brody, Kevin, Sonda i Ella Donovan, Chester i Shannon Andersen, Joe Lordi, Dave i Brenda Bartlett, porodica Rentschler, Lynn i Todd i njihove porodice. I, naravno, moji mama i tata.