

Predgovor

Ako ste čitali prethodne verzije ove knjige, uverićete se da je ovo izdanje najkorisnije i najdetaljnije dosad. Ukoliko vam trebaju brzi a detaljni odgovori o C++-u, korisniju knjigu nećete naći. Ona je ovoga puta poboljšana na dva načina: izdavač je izmenio format, a autor je obuhvatio nove karakteristike jezika.

Novi i bolji format

U novom milenijumu, knjige se ne takmiče samo međusobno već i s Internetom. Zato je za priručnike bitan lak pristup izloženoj građi, tj. brzina kojom možete da nađete ono što tražite.

Jedna od vrlina biblioteke „*Jasnim jezikom*“ jeste to što se knjige mogu pretražiti na više načina. Mi smo ovu vrlinu dodatno razvili.

- Zadatke možete potražiti u odeljku „C++ jasnim jezikom“, u prvom delu knjige. Umesto da budete upućeni samo na odgovarajući element jezika, bićete upućeni i na stranu na kojoj je on opisan.
- Pojmove možete da tražite po njihovom mestu u prvom, referentnom delu knjige, čija su zaglavlja poput onih u rečnicima. Pošto su odrednice navedene abecednim redom olakšano vam je kretanje kroz ovaj deo knjige.
- U svim referencama navedene su i stranice, tako da brže možete naći ono što tražite.

Ako ste čitali pređašnja izdanja ove knjige, uočićete još jednu bitnu promenu: lekcije su sada u drugom delu. Ova promena je u skladu s glavnim ciljem knjige: što više olakšati pristup informacijama. Odeljak orijentisan na zadatke, „C++ jasnim jezikom“, sada je na početku knjige, pa ćete još lakše naći ono što vam treba.

Informacija je beskorisna ako su objašnjenja loša. Nije bitno samo koliko brzo negde stignete, nego i da li vam pomaže to do čega ste došli. Da bih poboljšao kvalitet informacija, naveo sam bar po jedan primer za svaku temu, a gde je to bilo prikladno, dodao sam i razjašnjenje.

Čitaoci prethodnih izdanja će uvideti da je knjiga kvalitetnija nego pre. Dodao sam primere tamo gde su nedostajali. Osim toga, svaka tema ima odeljke „Namena“, „Sintaksa“ i „Primer“. Koristeći knjigu u ovom, novom formatu, shvatio sam da teme sada deluju jasnije, povezanije i preglednije.

Proširen opis novog C++-a

Prednost C++-a je u tome što postoji standard prihvaćen širom sveta: ANSI specifikacija. S vremena na vreme, komitet ANSI se sastaje i razmatra promene, pa je specifikacija, na sreću, poslednjih nekoliko godina bila stabilna.

Nekih promena je ipak bilo, usled sve veće važnosti internacionalnih aplikacija i činjenice da živimo u svetu astronomskih povećanja kapaciteta i memorije.

Dopuna 1

Ovaj dodatak ANSI specifikaciji predložen je sredinom devedesetih i sada je svuda prihvaćen, a uključuje korišćenje formata širokih znakova (`wchar_t`). Ovaj tip je neophodan zbog jezika koji imaju više od 255

tekstualnih znakova. Sam tip nije teško dokumentovati, ali Dopuna 1 istovremeno definiše veliki skup funkcija, kao što su `wprintf` i `wscanf`, kao podršku operacijama sa širokim znakovnim nizovima. U ovoj knjizi su opisane sve te funkcije.

C99

Ovo je novija promena i njeni efekti su manji od učinka Dopune 1. ISO C, specifikacija jezika C, sada predviđa dva nova prosta tipa koji koriste brže procesore i povećanu memoriju: `long long int` i `long double`. Većina proizvođača prevodilaca C++-a uključuje ove dodatke jezika C u svoje verzije C++-a. To je i logično, jer je C++ nastao kao poboljšana i proširena verzija C-a, s nekoliko ograničenja. U ovoj knjizi su opisana i ta dva nova tipa.

Klase ulazno-izlaznih tokova

Klase ulazno-izlaznih tokova su oduvek bile važan deo standardne biblioteke C++-a. Kao priznanje njihovoj sve većoj važnosti, u ovoj knjizi su potpunije i sistematičnije opisane te klase zajedno s funkcijama i objektima.

Kome je namenjena ova knjiga

Iako pruža dobru osnovu novajlijama, ova knjiga je namenjena i iskusnijim C++ programerima. Ona je korisna kao dopuna kursu ili drugom pažljivo vođenom učenju. Verujem da biste mogli naučiti C++ i samo iz ove knjige, ali lekcije su ovde sažetije. Ako ste početnik, dobro bi bilo da koristite još neku knjigu.

Deo I: Referentni pregled jezika C++

Deo I je predviđen za korisnike koji imaju osnovna znanja o C++-u i žele da brzo dobiju odgovore. Ako znate koji element jezika (rezervisanu reč, funkciju, klasu ili operator) treba da potražite, pronaći ćete ga u odeljcima 3–7. Na primer, sve rezervisane reči su u jednom odeljku, funkcije u drugom itd, pa je uglavnom očigledno gde šta treba da tražite. Ako ste u dilemi, u odeljku 2 „Spisak elemenata jezika C++“, naći ćete strane na kojima se određeni elementi nalaze. Oznake zaglavlja su slične onima u rečnicima, te je i time olakšano snalaženje.

Ako znate šta želite da uradite ali ne znate ime određene funkcije ili rezervisane reči, počnite traženjem zadatka u odeljku 1, „C++ jasnim jezikom“. Sada je to prvi odeljak knjige. U njemu je navedeno i ono što vam treba i tačno mesto gde to možete naći. Ovakva uputstva su od velike pomoći u C++-u, jer je on nasledio od C-a kompaktnija ali ponekad nejasnija imena funkcija, kao što je, npr, `strftime`.

Ako želite da se na ekranu ispiše dan u nedelji, a ne znate da treba da upotrebite funkciju `strftime`, u odeljku „C++ jasnim jezikom“ naći ćete `strftime` pod „dan u nedelji“, „mesec“, „vremenski podaci“ ili „vreme“.

Ova knjiga vam i na druge načine pomaže u vezi s funkcijama nejasnih imena. Za svaku stavku dat je i jasan opis, koji se nalazi u istom redu s elementom, i sažet iskaz o njegovoj nameni.

Deo II: C++ u praksi

Ako želite da počnete od pregleda objektno orijentisanog programiranja i programiranja u C++-u, pređite na deo II, „C++ u praksi“, na strani 302.

Drugi deo ima još jednu namenu, i zbog toga je premešten bliže kraju knjige. Pošto pronađete element jezika u delu I, možda će vam zatrebati još neke informacije o njemu. Na primer, posle čitanja o rezervisanoj reči `class` u odeljku 4, možda ćete želeti da pročitate nešto više o teoriji i upotrebi klasa; ova tema počinje u poglavlju 5. Prvi deo sadrži reference na drugi deo gde god to može da pomogne. U drugom delu ćete naći proširene opise i primere – neki se nalaze i u više poglavlja.

Nadam se da onima koji bi radije počeli od lekcija ne smeta što smo ih premestili u drugi deo knjige.

Rečnik

Rečnik koji je na kraju knjige popunjava konceptualne praznine i opisuje karakteristike jezika koje nisu povezane sa specifičnim rezervisanim rečima. Primera radi, tu ćete naći odrednice kao što su *functions* (funkcije), *pointer arithmetic* (pokazivačka aritmetika), *arrays* (nizovi) i *inheritance* (nasleđivanje). U njima vas, gde god to može da pomogne, upućujemo na delove I i II.

Rečnik ima još jednu važnu ulogu: da prevede žargon objektno orijentisanog programiranja na razumljiv jezik. Kada saznate pravo značenje reči kao što su *polimorfizam* ili *apstrakcija*, nećete više strahovati od susreta sa stručnjacima za objektno orijentisano programiranje i C++.

Konvencije primenjene u knjizi

C++ ponekad ima komplikovanu sintaksu, što odslikava moć i fleksibilnost jezika. Da bismo rezimirali brojne mogućnosti, u ovoj knjizi se koriste određene tipografske konvencije. Iako preterano korišćenje takvih konvencija može da komplikuje život, ja sam ih koristio gde god sam mislio da pojednostavljuju objašnjenja.

Najvažnija konvencija je da sve što je ispisano **polucрно** treba da se napiše upravo tako. Sve što je napisano *kurzivom* i neproporcionalnim fontom treba da zamenite nečim što sami izaberete. U sledećem primeru *tip* je ime bilo kog tipa u C++-u. Ovu sintaksu možete primeniti korišćenjem `sizeof(int)`, `sizeof(long)` ili `sizeof(float)`:

```
sizeof(tip)
```

Uglaste zagrade se povremeno koriste da bi pokazale opciju. U narednoj sintaksi zagrade označavaju da „C“ može biti uključeno ili izostavljeno, ali ih ne unosite:

```
extern ["C"] deklaracija;
```

Ako je predviđeno da se napišu i zagrade, onda su one odštampane polucрно. To je slučaj s nizovima i operatorima `new` i `delete`. Na primer, da biste koristili sledeću sintaksu, možete uneti `new int[100]` ili `new long[56]`:

```
new tip [duzina]
```

U odeljku 6 nalaze se deklaracije svih funkcija. U deklaraciji su navedeni svaki argument i njegov tip. Kada koristite funkciju, nikada ne unosite tip već samo argument. Na primer, sledeća deklaracija znači da `abs` dobija i vraća celobrojnu vrednost (`int`):

```
int abs(int broj);
```

Evo primera pozivanja ove funkcije:

```
int i = abs(-75);
```

Ostale oznake

U ovoj knjizi ima nekoliko tipova oznaka koje treba da privuku pažnju na raznovrsna obaveštenja.

● ANSI

Ova oznaka ističe karakteristiku jezika u ANSI specifikaciji koje nema u prvim verzijama C++-a.

● **NAPOMENA**

Ovo je oznaka tehničkog objašnjenja koje odstupa od glavne teme. U C++-u se često pojavljuju specijalni slučajevi i dopune. Da bi sve bilo tehnički korektno, a da vas pri tome ne ometa previše, tekst u vidu napomene pruža dodatne informacije.

● **C/C++**

Ova oznaka ističe informacije o uobičajenoj karakteristici jezika koja postoji i u C-u i u C++-u ali ima različito značenje. Jezik C ima neka ograničenja koja C++ nema i obratno. Obratite naročitu pažnju na ova obaveštenja ako prebacujete C kôd u C++, ili ste, pak, programer u C-u a počinjete da programirate u C++-u.

● **SAVET**

U tekstu se preporučuje alternativni način da nešto uradite brže ili prikladnije. Saveti su uvek u obliku predloga.

Zašto C++?

Uz Visual Basic i Javu, C++ je jezik koji obavezno treba da znate ako nameravate da se u budućnosti bavite programiranjem. Od ova tri jezika, C++ omogućuje najpotpuniju realizaciju objektno orijentisanosti i najpogodniji je za pisanje profesionalnih, komercijalnih aplikacija i za sistemsko programiranje. Ukratko, temeljno poznavanje C++-a vas svrstava u prvu ligu programera.

Prepreka u učenju C++-a je što su imena funkcija kompaktna i često na osnovu njih ne možete odmah prepoznati namenu funkcije. Ova knjiga vam priskazuje u pomoć jasnim jezikom i predočava sadržaj organizovan po zadacima, kao i sažetak svake funkcije i rezervisane reči.

Dodatna prepreka je što se C++ oslanja na operatorske simbole gde bi, recimo u Basicu, bila upotrebljena engleska reč (na primer, operacija AND). Ova knjiga vam i tu priskazuje u pomoć, s nekoliko sažetih objašnjenja operatora i njihove upotrebe.

Završna reč autora

C++ je veoma elegantan i moćan jezik. Međutim, mnogi mu nisu naklonjeni, pre svega zbog činjenice da je nastao iz C-a, starijeg i manje moćnog jezika, pa je morao da zadrži stare karakteristike zbog kompatibilnosti. Isto tako, mana mu je što ne pravi ustupke početnicima u programiranju,

kojima su teži pojmovi kao što su pokazivači i reference. C i C++, možda nepravedno, imaju reputaciju „španskog sela“, iako je nekoliko miliona ljudi uspelo da ih nauči ili je bar pokušalo.

Nadam se da se nećete zbunjivati tokom rada u C++-u. Pokušao sam da vam prenesem ono što sam naučio iz sopstvenih uspeha i grešaka – grešaka koje ne bih napravio da mi je neko ukazao na redundancije i zamke za koje znam sada. Želim da vam pisanje programa bude ugodno, elegantno i bez grešaka.

Još jedno obaveštenje. Sve komentare možete poslati na adresu `Briano2u@aol.com`, a kôd za primere naći ćete na adresi:

<http://www.hungryminds.com/extras/cipe/>