Sadržaj

Uvod 1

I DEO: Početak

1. ČAS: Postanite programer 3
Biranje jezika 4
Kažite računaru šta da radi 6
Kako programi rade 7
Kada programi ne rade 8
Biranje programerske alatke za Javu 9
Instaliranje programerske alatke za Javu 10

2. ČAS: Pisanje prvog programa 13
Šta vam treba za pisanje programa 13
Pravljenje programa `Saluton` 14
Započinjanje programa 14
Iskaz `class` 16
Šta radi iskaz `main` 17
Vitičaste zagrade 17
Spremanje informacija u promenljivu 18
Prikazivanje sadržaja promenljive 18
Sačuvajte završeni proizvod 19
Prevođenje programa u jedan Class fajl 20
Ispravljanje grešaka 20
Izvršavanje Java programa 22

3. ČAS: Odmaranje u Javi 27
Prva stanica: Oracle 27
Kratka istorija Jave 29
U školu sa Javom 30
Ručak u Food Networku 32
Posmatranje neba kod NASA-e 33
Prelazak na posao 34
Stanica SourceForge za dobijanje pravca 35

4. ČAS: Kako rade Java programi 41
Pravljenje aplikacije 41
Slanje argumenata u aplikacije 43
Biblioteka Java Class Library 45
Isprobavanje Java iskaza u ljusci JShell 49

II DEO: Učenje osnova programiranja

5. ČAS: Spremanje i menjanje informacija u programu 53
Iskazi i izrazi 53
Određivanje tipa promenljive 54
Celi brojevi i brojevi u pokretnom zarezu 54
Znakovi i stringovi 55
Drugi tipovi numeričkih promenljivih 56
Tip promenljive `boolean` 57
Imenovanje promenljivih 58
Spremanje informacija u promenljive 59
Sve o operatorima 60
Povećavanje i umanjivanje promenljive 61
Prioritet operatora 63
Upotreba izraza 64

6. ČAS: Upotreba stringova za komunikaciju 71
Smeštanje teksta u stringove 71
Prikazivanje stringova u programima 72
Specijalni znakovi u stringovima 73
Spajanje stringova 74
Upotreba promenljivih sa stringovima 74
Napredna obrada stringova 75
Poređenje dva stringa 76
Utvrđivanje dužine stringa 77
Kopiranje stringa uz promenu veličine slova 77

Traženje stringa 77
Prikaz špice 78

7. ČAS: Uslovna ispitivanja za donošenje odluka 83
Iskazi `if` 84
Poređenja manje od i veće od 84
Poređenja jednako i nejednako 85
Organizovanje programa pomoću blok iskaza 85
Iskazi `if-else` 87
Iskazi `switch` 87
Ternarni operator 90
Posmatranje sata 91

8. ČAS: Ponavljanje postupaka u petlji 99
`for` petlje 99
`while` petlje 102
`do-while` petlje 103
Napuštanje petlje 104
Imenovanje petlje 105
Složene `for` petlje 106
Testiranje brzine vašeg računara 107

III DEO: Rad sa informacijama na nove načine

9. ČAS: Spremanje informacija u nizove 111
Pravljenje nizova 112
Upotreba nizova 113
Višedimenzionalni nizovi 115
Sortiranje niza 116
Brojanje znakova u stringu 118

10. ČAS: Pravljenje prvog objekta 125
Kako funkcioniše objektno orijentisano programiranje 125
Objekti na delu 127
Šta su objekti 128
Nasleđivanje 129
Izgradnja hijerarhije nasleđivanja 130
Konvertovanje objekata i jednostavnih promenljivih 131
Konvertovanje jednostavnih promenljivih 131
Konvertovanje objekata 132
Konvertovanje jednostavnih promenljivih u objekte i nazad 133
Pravljenje objekta 135

11. ČAS: Opisivanje objekta 141
Pravljenje promenljivih 141
Napravite promenljivu klase 143
Pravljenje ponašanja sa metodima 144
Deklarisanje metoda 145
Slični metodi sa različitim argumentima 146
Konstruktori 147
Metodi klase 148
Opseg promenljivih unutar metoda 148
Stavljanje jedne klase u drugu klasu 150
Ključna reč `this` 151
Upotreba metoda i promenljivih klase 152

12. ČAS: Iskorišćavanje postojećih objekata 159
Moć nasleđivanja 159
Nasleđivanje ponašanja i atributa 160
Nadjačavanje metoda 161
Uspostavljanje nasleđivanja 161
Ključne reči `this` i `super` u potklasi 162
Rad sa postojećim objektima 163
Spremanje objekata iste klase u ArrayList 164
Petlja po ArrayListi 166
Pravljenje potklase 168

IV DEO: Prelazak na napredne teme

13. ČAS: Čuvanje objekata u strukturama podataka 173
ArrayListe 174
HashMape 180

14. ČAS: Obrada grešaka u programu 187
Izuzeci 188
Hvatanje izuzetaka u bloku `try-catch` 189
Hvatanje više različitih izuzetaka 192
Raditi nešto posle izuzetka 194
Bacanje izuzetaka 194
Zanemarivanje izuzetaka 196
Izuzeci za koje nije potreban `catch` 197
Bacanje i hvatanje izuzetaka 197

15. ČAS: Pravljenje programa sa nitima 203
Niti 203
Usporavanje programa 204
Pravljenje niti 204
Rad sa nitima 208
Deklaracija `class` 209
Postavljanje promenljivih 209
Konstruktor 210
Hvatanje grešaka dok se postavljaju URL-ovi 211
Pokretanje niti 211
Izvršavanje niti 212
Obrada pritiskanja miša 213
Prikazivanje linkova koji se vrte 213
Zaustavljanje niti 216

16. ČAS: Upotreba unutrašnjih klasa i zatvoreni opsezi 221
Unutrašnje klase 222
Anonimne unutrašnje klase 225
Zatvoreni opsezi 229

V DEO: Programiranje grafičkog korisničkog interfejsa

17. ČAS: Pravljenje jednostavnog korisničkog interfejsa 237
Swing i komplet alatki Abstract Windowing Toolkit 237
Komponente 238
Prozori i okviri 238
Dugmad 242
Natpisi i tekstualna polja 245
Polja za potvrdu 246
Kombinovani okviri 247
Tekstualna područja 248
Paneli 251
Napravite vlastitu komponentu 251

18. ČAS: Planiranje korisničkog interfejsa 259
Menadžeri rasporeda 259
Menadžer `GridLayout` 261
Menadžer `BorderLayout` 262
Menadžer `BoxLayout` 263
Razdvajanje komponenti pomoću `Insets` 264
Raspoređivanje aplikacije 265

19. ČAS: Reagovanje na ulaz od korisnika 273
Postizanje da vaš program sluša 273
Postavljanje komponenti koje treba da se čuju 274
Postupanje sa korisničkim događajima 275
Događaji polja za potvrdu i kombinovanih okvira 276
Događaji tastature 276
Omogućavanje i onemogućavanje komponenti 278
Kompletiranje grafičke aplikacije 279

VI DEO: Pisanje aplikacija za Internet

20. ČAS: Čitanje i pisanje fajlova 291
Tokovi 291
Fajlovi 292
Čitanje podataka iz toka 293
Baferovani ulazni tokovi 297
Pisanje podataka u tok 299
Čitanje i pisanje svojstava konfiguracije 302

21. ČAS: Novi HTTP klijent u Javi 9 309
Java moduli 309
HTTP zahtev 310
Čuvanje fajla sa Veba 314
Slanje podataka na Veb 317

22. ČAS: Pravljenje Java2D grafike 325
Klasa `Font` 325
Klasa `Color` 326
Pravljenje prilagođenih boja 327
Crtanje linija i oblika 328
Crtanje linija 328
Crtanje pravougaonika 329
Crtanje elipsi i krugova 330
Crtanje lukova 330
Kružni dijagram 331

23. ČAS: Pravljenje Minecraft modifikacija u Javi 341
Postavljanje Minecraft servera 342
Rešavanje problema sa izvršavanjem servera 344
Povezivanje sa serverom 346
Rešavanje problema povezivanja sa serverom 347
Pravite svoju prvu modifikaciju 348
Naučite zombije da jašu konje 356
Pronađite sve čopore (i ubijte ih) 362
Pisanje modifikacije koja može da pravi stvari 366

24. ČAS: Pisanje Android aplikacija 375
Uvod u Android 375
Pravljenje Android aplikacije 377
Pretraživanje novog Android projekta 378
Pravljenje aplikacije 380
Postavljanje Android emulatora 382
Izvršavanje aplikacije 383
Projektovanje prave aplikacije 385
Organizovanje resursa 386
Konfigurisanje manifesta aplikacije 387
Projektovanje korisničkog interfejsa 388
Pisanje Java koda 390

DODACI

DODATAK A: Upotreba IDE okruženja NetBeans 399
Instaliranje NetBeansa 399
Pravljenje novog projekta 400
Pravljenje nove Java klase 402
Izvršavanje aplikacije 404
Rešavanje grešaka 404

DODATAK B: Kuda odavde: Java resursi 407
Druge knjige 407
Zvaničan Oracle Java sajt 408
Dokumentacija Java klasa 408
Drugi Java veb sajtovi 408
Zvaničan sajt ove knjige 408
Workbench 409
Slashdot 409
Drugi blogovi o Javi 409
InformIT 409
Stack Overflow 409
Časopis *JavaWorld* 409
Java adresar na Developer.com 410
Java susreti 410
Mogućnosti za zapošljavanje 410

DODATAK C: Veb sajt ove knjige 411

DODATAK D: Rešavanje problema sa emulatorom Android Studija 413
Problemi sa izvršavanjem mobilnih aplikacija 413
Instaliranje HAXM-a u Android Studiju 414
Instaliranje HAXM-a na vaš računar 415
Proveravanje BIOS parametara 417

DODATAK E: Greška 'Package Not Visible' u NetBeansu 419
Dodavanje klase Module Info 419

Indeks 421

O autoru

Rogers Cadenhead je pisac, programer računara, i veb programer koji je napisao preko 25 knjiga o programiranju- i temema vezanim za Internet, uključujući *Naučite sami Javu za 21 Dan*. On održava sajt Drudge Retort i druge veb sajtove koji primaju preko 20 miliona poseta godišnje. Zvanični veb sajt ove knjige je www.java24hours.com, a piscu se može obratiti na Twitteru na @rcade.

Priznanja

Svima u Pearsonu – pogotovo Marku Taberu, Lori Lyons, Abigail Manheim Bass, Borisu Minkin i Dhayanidhi Karunanidhi. Nijedan pisac ne može sam da napravi ovakvu knjigu. Njihov izvrstan rad će mi obezbediti mnoge pohvale kasnije.

Mojoj ženi Mary i mojim sinovima Maxu, Eliju i Samu.

Uvod

Kao autor računarskih knjiga, provodim dosta vremena vrebajući po računarskim odeljcima knjižara i posmatrajući ponašanje čitalaca dok se pretvaram da čitam najnovije izdanje časopisa o sapunskim serijama.

Zahvaljujući istraživanju, znam da ako ste uzeli ovu knjigu i pogledali ovaj uvod, meni ostaje još samo 13 sekundi pre nego što je spustite i pređete za šank da naručite late dupli-produženi-bez-kofeina-sa-duplo-vanile-dobro-promešajte.

Zato ću biti kratak: programiranje računara u jeziku Java je lakše nego što izgleda.

Ne bih smeo to da vam kažem zato što je na hiljade programera upotrebilo svoje Java veštine da dobije dobro plaćene poslove u razvijanju softvera, programiranju servera i izradi Android aplikacija. Nijedan programer nikako ne želi da gazda sazna da svako ko je uporan i ima malo slobodnog vremena može da nauči ovaj jezik, najpopularniji programski jezik na planeti. Krčenjem puta kroz svako od jednosatnih poglavlja knjige *Java za 24 časa – Naučite sami*, moći ćete brzo da naučite programiranje u jeziku Java.

Svako može da nauči da piše računarske programe, čak i onaj ko ne ume da programira video rekorder. Java je jedan od najboljih programskih jezika koje možete da naučite zato što je to korisna, moćna, savremena tehnologija koju su prihvatile kompanije u celom svetu.

Ova knjiga je namenjena onima koji nisu programeri, novim programerima koji misle da mrze programiranje i iskusnim programerima koji žele da što pre postignu ubrzanje sa Javom. Koristi se Java 9, najnovija i najmoćnija verzija ovog jezika.

Java je izuzetno popularan programski jezik zbog svega što omogućava. Možete da pravite programe koji imaju grafički korisnički interfejs, koji se povezuju sa veb servisima, izvršavaju se na Android telefonu ili tabletu i još mnogo toga.

Ovaj jezik se može naći na neverovatnim mestima. Jedno od njih je Minecraft, igrački fenomen napisan isključivo u Javi. (U ovoj knjizi učite kako da pravite Java programe koji se izvršavaju u toj igri, pored gmizavaca i zombi pigmenta!)

U ovoj knjizi se Java programiranje podučava od nule. Koncepti se uvode govornim jezikom i na engleskom, a ne stručnim žargonom, a primere pravih programa ćete praviti korak po korak. Provedite 24 časa sa ovom knjigom, pa ćete pisati vlastite Java programe,

sigurni u svoju sposobnost da koristite jezik i učite više o njemu. Takođe ćete steći veštine koje postaju sve važnije – kao što je Internet računarstvo, dizajn grafičkih korisničkih interfejsa, pravljenje aplikacija i objektno orijentisano programiranje.

Ti izrazi vam trenutno možda ne znače mnogo. U stvari, to su verovatno stvari zbog kojih programiranje izgleda zastrašujuće i teško. Međutim, ako umete na računaru da napravite album fotografija za Facebook, plaćate porez, ili radite u Excelovom radnom listu, moći ćete da naučite pisanje računarskih programa čitanjem knjige *Java za 24 časa – Naučite sami*.

NAPOMENA

Sada, ako ste pre za kafu nego za Javu, molim da ovu knjigu vratite na policu tako da naslovna strana bude vidljiva na kraju police gde prolazi što više posetilaca prodavnice.

1. čas
Postanite programer

Spisak aktivnosti za ovaj čas:

- Saznajete razlog za učenje Jave.
- Otkrivate kako programi rade.
- Odabirate alatku za programiranje Jave.
- Pripremate se za pisanje svog prvog programa.

Verovatno ste čuli da je programiranje računara užasno teško. Potrebna je diploma računarskih nauka, na hiljade dolara za hardver i softver računara, oštro analitičko razmišljanje, strpljenje jednog Jova i velika sklonost kafeinskim pićima.

Osim onog što se tiče kafeina, niste dobro čuli. Programiranje je lakše nego što biste pomislili, uprkos onome što programeri godinama pričaju, samo da bismo lakše dolazili do visoko plaćenih poslova.

Ovo je odlično vreme da se uči programiranje. Bezbroj programerskih alatki postaje raspoloživo na Vebu za besplatno preuzimanje, a na hiljade programera deli svoj rad kao otvoreni kôd da drugi ljudi mogu da ispitaju kako je softver napisan, isprave greške i dodaju unapređenja. U privredi koja se oporavlja, mnoge kompanije zapošljavaju programere.

Ovo je odlično vreme da se uči Java, zato što se taj jezik nalazi svuda. Milijarde mobilnih uređaja koristi Android, operativni sistem čije su sve aplikacije pisane u Javi. Ako imate Android telefon, uživali ste u radu Java programera kad god ste gledali film, ljuljali se uz radio koji se preuzima u realnom vremenu, ili pogodili neprijateljskog letećeg stvora u loše izgrađenom zamku *Swine Fortress*.

Ova knjiga ima za cilj da tri vrste ljudi nauči Java programiranju:

1. Nervozne početnike koji nikad ranije nisu pokušali da programiraju
2. Ogorčene početnike koji su pokušali da programiraju ali im je to bilo mrsko kao što je Lordu Voldemoru mrska siročad u britanskim školama
3. Nestrpljive intelektualce koji znaju neki drugi programski jezik i hoće što pre da ubrzaju stvari u jeziku Java

Da bi se postigao taj cilj, u ovoj knjizi se koliko je god moguće koristi govorni jezik umesto tehničkog žargona ili nerazumljivih akronima. Svi novi programerski pojmovi se temeljito objašnjavaju čim se uvedu.

Ako sam uspeo, kad završite ovu knjigu imaćete dovoljno programerskih veština da predstavljate opasnost sebi i drugima. Bićete u stanju da pišete programe, smelije se upuštate u programske klase i knjige i lakše učite nove jezike. (Da budemo jasni, programske jezike. Ova knjiga vam neće pomoći da savladate španski, esperanto, ili klingonski.)

Bićete takođe vešti u Javi, najčešće korišćenom programskom jeziku na planeti.

Prvi čas u ovoj knjizi je uvod u programiranje i uputstvo za podešavanje vašeg računara tako da ga možete koristiti za pisanje i izvršavanje Java programa.

Biranje jezika

Ako vam nije problem da na računaru pripremite biografiju koja lepo izgleda, saldirate čekovnu knjižicu, ili podelite svoje fotografije sa letovanja na Instagramu, moći ćete da pravite računarski softver.

Ključno za učenje programiranja je da počnete od pravog jezika. Programski jezik koji izaberete često zavisi od zadataka koje hoćete da obavite. Svaki jezik ima svoje prednosti i mane. U moje vreme mladih šminkera, ljudi su učili programiranje u jeziku BASIC zato što je on napravljen za početnike.

NAPOMENA

Jezik BASIC je izmišljen da bi ga studenti lako naučili (slovo B u reči BASIC znači Beginner – početnik). Loša strana jezika BASIC je da se sa tim jezikom lako stekne navika neurednog programiranja.

Najpopularniji jezik koji danas koristi BASIC je Visual Basic, programski jezik iz Microsofta koji je daleko prevazišao svoje korene. VB, kako se još naziva, projektovan je za pravljenje programa koji će raditi na računarima i mobilnim uređajima koji koriste operativni sistem Windows. Drugi popularan jezik je PHP, jezik za skriptovanje kojim se prave veb sajtovi. Još neki jezici u širokoj upotrebi za koje ste možda čuli su C++, Ruby, Javascript i Python.

Svaki od ovih jezika ima svoje pristalice, ali na časovima informatike u srednjoj školi kao i na fakultetskom nivou najčešće se uči Java.

Programski jezik Java, koji nudi Oracle, teže se uči od nekih drugih jezika kao što su VB i PHP, ali to je dobra početna tačka iz nekoliko razloga. Jedna prednost učenja Jave je da možete da ga koristite na raznovrsnim operativnim sistemima i u raznovrsnim računarskim okruženjima. Java programi mogu da budu softver za stone računare, veb aplikacije, veb servere, Android aplikacije i tako dalje, i da se izvršavaju na operativnim sistemima

Windows, Mac, Linux i drugim. Ta prilagodljivost se pominje u ambicioznom ranom Java sloganu „Napišite jednom, izvršite bilo gde".

NAPOMENA

Rani Java programeri su imali manje laskav slogan: „Napišite jednom, tražite greške bilo gde". Jezik je prešao dug put, pošto je prva verzija izdata još 1996.

Još jedna značajna prednost je to da Java zahteva veoma organizovan pristup da bi programi funkcionisali. Morate biti veoma brižljivi u tome kako pišete programe i u tome kako oni spremaju i menjaju podatke.

Kada počnete da pišete Java programe, možda nećete sitničarsko ponašanje ovog jezika doživeti kao prednost. Možda će vam biti dosadno da pišete program i da morate da ispravljate više grešaka da bi program uopšte mogao da se izvrši. Prednost tog dodatnog napora je da će softver koji napravite biti pouzdaniji, korisniji i da neće imati greške.

Na sledećim časovima, naučićete sva pravila Jave i sve zamke koje treba izbeći.

Javu je izmislio kanadski računarski naučnik James Gosling kao bolji način da se prave programi za računare. Dok je radio u firmi Sun Microsystems godine 1991, Gosling nije bio zadovoljan performansama programskog jezika C++ na jednom projektu, pa je napravio nov jezik koji je bolje obavio posao. Da li je Java superiorniji programski jezik od drugih, to je, naravno, tema žestokih debata, ali uspeh ovog jezika dokazuje snagu njegovog inicijalnog dizajna. Petnaest milijardi uređaja na celom svetu izvršava Javu, broj je toliko zapanjujući da ću ga ponoviti. Petnaest milijardi! Od kad je uveden, izdato je preko 1.000 knjiga o ovom jeziku. (Meni je ovo dvadeseta.)

Bez obzira na to da li je Java najbolji jezik, svakako je odlično da se on uči. Prvu priliku da isprobate Javu imaćete na 2. času, „Pisanje prvog programa".

Kad naučite jedan programski jezik, mnogo je lakše učiti sledeće jezike.

Mnogi su međusobno slični, pa ne počinjete od nule kada krenete u nov. Na primer, mnogi C++ i Smalltalk programeri smatraju da je prilično lako naučiti Javu zato što Java koristi zamisli iz tih starijih jezika. Slično tome, C# prihvata mnoge ideje iz Jave, pa je Java programerima lakše da ga prihvate.

NAPOMENA

C++ se na ovom času pominje nekoliko puta, pa možda zastajete na tom izrazu i pitate se šta on znači – i kako se izgovara. C++ se čita „C-Plus-Plus", a to je programski jezik koji je razvio danski računarski naučnik Bjarne Stroustrop iz Bell Laboratories. C++ je unapređenje programskog jezika C, pa otud deo naziva „Plus-Plus". Zašto ga nije prosto nazvao C+? To Plus-Plus je programerska šala koju ćete razumeti kasnije u ovoj knjizi.

Kažite računaru šta da radi

Računarski program, koji se zove i softver, je način da se računaru kaže da izvrši neki zadatak. Sve što računar radi, od pokretanja do gašenja, radi neki program. Mac OS X je program; Minecraft je program; drajverski softver koji kontroliše vaš štampač je program; čak i užasavajući plavi ekran smrti na Windows PC-ju koji je otkazao je takođe program.

Računarski programi se sastoje od liste komandi koje računar obrađuje konkretnim redom kada se program izvršava. Svaka komanda se zove *iskaz* (engl. *statement*).

Kad biste u kući imali batlera i da ste vi čudak opsednut kontrolom, ličnost tipa A, vi biste svom slugi zadali detaljna uputstva šta mora da izvrši svaki dan, ovako:

> Dragi g. Dživs,
>
> Molim vas da obavite za mene sledeće poslove dok ja tražim pomoć od kongresa:
>
> Tačka 1: Usisajte dnevnu sobu.
>
> Tačka 2: Idite u prodavnicu.
>
> Tačka 3: Uzmite soja sos, wasabi i koliko god kalifornijskih suši rolnica možete da ponesete
>
> Tačka 4: Vratite se kući.
>
> Iskreno, vaš poslodavac i gazda,
>
> Bertie Wooster

Ako živom batleru kažete šta da uradi, vaši zahtevi bi mogli da se izvrše uz izvesna odstupanja. Ako nema kalifornijskih rolnica, Dživs bi mogao da donese bostonske rolnice.

Kod računara nema odstupanja. Oni uputstva izvršavaju doslovno. Programi koje pišete se izvršavaju precizno, instrukcija po instrukcija.

Sledeći primer je računarski program od tri reda, napisan u jeziku BASIC. Pogledajte ga, ali ne razmišljajte previše o tome šta treba svaki red da znači.

```
1 PRINT „Hej Tome, ovde Bob iz kancelarije na kraju hodnika.”
2 PRINT „Radujem se da te vidim, druškane. Kako je bilo?”
3 INPUT A$
```

Prevedeno na govorni jezik, ovaj program je kao da ste računaru zadali sledeći spisak aktivnosti:

> Dragi personalni računaru,
>
> Tačka 1: Prikaži poruku, „Hej Tome, ovde Bob iz kancelarije na kraju hodnika”.

Tačka 2: Postavi pitanje, „Radujem se da te vidim, druškane. Kako je bilo?"

Tačka 3: Pruži korisniku priliku da odgovori na pitanje.

Iskreno, tvoj poslodavac i gazda,

Ima Koder

Svaki red u računarskom programu se zove *iskaz*. Računar obrađuje svaki iskaz u programu konkretnim redom, isto kao što kuvar prati recept ili batler g. Dživs izvršava naredbe Bertija Woostera. U jeziku BASIC, brojevi redova se koriste da bi se iskazi postavili u pravilan redosled. Drugi jezici, kao što je Java ne koriste brojeve redova, već koriste drugačije načine da bi saopštili računaru kako da izvrši program.

Zbog načina na koji programi funkcionišu, vi ne možete da krivite računar ako nešto nije u redu kada se vaš program izvršava. Računar radi tačno ono što ste mu vi rekli da uradi, pa je krivica za greške obično na programeru.

To je loša strana. Dobra strana je da ne možete da napravite nikakvu trajnu štetu. Računar se neće oštetiti dok vi budete učili da programirate u jeziku Java.

Kako programi rade

Kolekcija iskaza koji čine računarski program je njegov *izvorni kôd* (engl. *source code*).

Većina računarskih programa se piše na isti način na koji pišete e-poštu – iskaze upisujete u tekstualni prozor. Neke alatke za programiranje imaju vlastiti editor izvornog koda, a neke se koriste sa bilo kojim softverom za uređivanje teksta.

Kada završite pisanje računarskog programa, vi sačuvate fajl na disk. Računarski programi često imaju vlastitu ekstenziju za ime fajla, koja označava koji je to tip fajla. Java programi moraju da imaju ekstenziju `.java`, kao u `Calculator.java`.

NAPOMENA

Računarske programe treba pripremiti kao tekstualne fajlove bez specijalnog formatiranja. Notepad, jedan tekst editor koji se isporučuje uz Windows, sprema sve fajlove kao neformatirani tekst. Za pravljenje tekstualnih fajlova bez formatiranja možete takođe da koristite TextEdit na Macu ili vi editor ili emacs na Linux sistemima. Kasnije na ovom času ćete videti jedno jednostavnije rešenje.

Da biste izvršili program koji ste sačuvali kao fajl, treba vam neka pomoć. Vrsta potrebne pomoći zavisi od programskog jezika koji koristite. Za neke jezike je potreban interpreter koji izvršava programe. Interpreter ispituje svaki red računarskog programa i izvrši taj red, zatim prelazi na sledeći red. Mnoge verzije jezika BASIC su interpretirani jezici.

Najveća prednost interpretiranih jezika je to što se oni brže testiraju. Kada pišete program u jeziku BASIC, možete odmah da ga isprobate, ispravite greške,i pokušate ponovo. Osnovni nedostatak je to da se interpretirani jezici izvršavaju sporije od drugih programa. Svaki red mora da se prevede u instrukcije koje računar može da izvrši, red po red.

Drugi programski jezici zahtevaju prevodioca (engl. *compiler*). Prevodilac uzme program i prevede ga u jedan oblik koji računar razume. On takođe postiže da se program izvršava što je moguće efikasnije. Prevedeni program može direktno da se izvršava bez potrebe za interpreterom.

Prevedeni programi se izvršavaju brže od interpretiranih programa, ali je za testiranje potrebno više vremena. Morate da napišete svoj program i prevedete ga celog da biste ga isprobali. Ako otkrijete neku grešku i ispravite je, morate ponovo da prevedete program.

Java je neobična jer joj je potreban i prevodilac i interpreter. Iskaze od kojih se program sastoji prevodilac konvertuje u bajtkôd. Pošto se ovaj bajtkôd uspešno napravi, može da ga izvrši interpreter koji se zove Java Virtual Machine.

Java Virtual Machine, koja se naziva i JVM, je ono što omogućava da se isti Java program izvršava bez ikakve izmene na različitim operativnim sistemima i na različitim računarskim uređajima. Virtuelna mašina pretvara bajtkôd u instrukcije koje može da izvrši operativni sistem konkretnog uređaja.

NAPOMENA

Java 9 uvodi novu alatku po imenu JShell koja služi kao interpreter, pošto izvršava Java iskaz čim ga upišete. JShell radi tako što stavlja iskaz u Java program, prevodi taj program u bajtkôd i izvrši ga. To je korisna alatka za učenje i za testiranje.

Kada programi ne rade

Mnogi novi programeri se obeshrabre kad počnu da testiraju svoje programe. Greške se javljaju svuda. Neke od njih su sintaksne greške, koje računar uoči kada pregleda program pa ga zbuni način na koji je iskaz napisan. Ostale greške su logičke, koje primeti tek programer prilikom testiranja programa (ili mu one potpuno promaknu). Logičke greške često dovode do toga da program uradi nešto nepredviđeno.

Kad budete počeli da pišete programe, dobro ćete upoznati greške. One su prirodan deo procesa. Programske greške se zovu *bugs* (bube), izraz koji je star preko sto godina i opisuje greške u tehničkim uređajima.

Proces ispravljanja grešaka se na engleskom zove: *debugging*.

Nije čudo da se greške opisuju na toliko različitih načina. Dok budete učili programiranje, steći ćete mnogo iskustva u ispravljanju grešaka – hteli ne hteli.

NAPOMENA

Jednu od prvih računarskih „buba" – kako nazivaju greške – otkrio je 1947 jedan tim u kojem je bila američka računarska naučnica Grace Hopper. Hopper je testirala računar na Harvardu i jedan relej nije radio kako treba. Uzrok nije bio softverski problem – to je bila prava buba! Jedan član tima je rešio problem tako što je uklonio mrtvog moljca i zalepio ga u evidencionu knjigu uz napomenu, „Prvi pravi slučaj pronalaženja bube". Buba i strana dnevnika mogu se videti na www.doncio.navy.mil/CHIPS/ArticleDetails.aspx?id=3489.

Biranje programerske alatke za Javu

Da biste počeli da pišete Java programe, morate da imate alatku za programiranje Jave. Dostupno je više takvih programa za Javu, uključujući jednostavan JDK – Java Development Kit i složenije Eclipse, IntelliJ IDEA i NetBeans. Poslednje tri alatke su integrisana okruženja za razvoj (IDE – Integrated Development Environment), moćne alatke koje profesionalni programeri koriste u svom radu.

Kad god Oracle izda novu verziju Jave, prva alatka koja je podržava je JDK (Java Development Kit).

Da biste pravili programe iz ove knjige, moraćete da koristite JDK version 9 ili neku programersku alatku koja radi u njoj. JDK je skup besplatnih alatki sa komandne linije za pravljenje Java softvera. On nema grafički korisnički interfejs, pa ako nikad niste radili u okruženju bez grafičkog interfejsa, kao što je komandna linija u Windowsu ili Linuxu, neće vam biti lako da koristite JDK.

IDE okruženje NetBeans, koje Oracle takođe nudi besplatno je mnogo lakši način za pisanje i testiranje Java koda nego JDK. NetBeans sadrži grafički korisnički interfejs, editor izvornog koda, dizajner korisničkih interfejsa i menadžer projekta. Ovo okruženje radi zajedno sa JDK-om, tako što ga izvršava iza scene, pa morate u svom sistemu da imate obe ove alatke kada počnete da razvijate Java programe.

Većina programa u ovoj knjizi izrađena je u okruženju NetBeans, koje možete da preuzmete i instalirate odvojeno od JDK. Možete takođe da koristite i druge Java alatke pod uslovom da podržavaju JDK 9.

NAPOMENA

U ovoj knjizi ne morate da koristite NetBeans. Ako možete da koristite JDK ili neku drugu alatku da napravite, prevedete i izvršite program, većina projekata zahteva samo te zadatke. NetBeans se opisuje jer se za čitaoce prethodnih izdanja pokazalo da je lakši nego JDK. Ja koristim NetBeans za veći deo svog Java programiranja.

Na 24. času, „Pisanje Android aplikacija”, koristićete IDE okruženje Android Studio. Google, koji je napravio Android, preporučuje tu besplatnu alatku Java programerima koji prave programe za Android.

Instaliranje programerske alatke za Javu

Svaki čas u ovoj knjizi sadrži projekte Java programiranja koje možete da preduzmete da biste unapredili svoje znanje o materiji tako da se zadubite u nju.

Nećete moći da se zadubite u materiju ako u svom računaru nemate alatku za programiranje Jave.

Ako ste već instalirali neku alatku koja podržava Javu, možete da koristite nju da biste napravili udžbeničke programe za sledeća 23 časa. Međutim, trebalo bi da ste već donekle navikli na tu alatku. Istovremeno učiti Javu i neko složeno IDE okruženje moglo bi da se pokaže teško.

Za programiranje uz čitanje ove knjige preporučujem NetBeans, koji je besplatan na Oracleovom veb sajtu http://netbeans.org. Iako NetBeans ima napredne funkcije za čije učenje je potrebno vreme, sa njim se jednostavne Java aplikacije lako prave i izvršavaju.

Da biste videli kako se NetBeans preuzima i instalira, pročitajte Dodatak A, „Upotreba IDE okruženja NetBeans”.

Rezime

Na ovom času, predstavljen vam je koncept programiranja računara – računaru se daje skup instrukcija koje se zovu iskazi i oni mu govore šta da radi. Takođe ste naučili zašto je dobro što ste izabrali da sami naučite Javu, a ne neki drugi programski jezik.

Možda ste već preuzeli i instalirali neku alatku za programiranje Jave koju ćete koristiti za pravljenje programa na sledećim časovima.

Pitajte 10 programera koji je najbolji programski jezik i možda ćete dobiti 10 odgovora, uz izazove tipa „moj jezik je jači od tvog jezika” i viceve tipa „tvoj izvorni kôd je tako naduven”. Java u takvim raspravama dobro prolazi jer je široko prihvaćena, krajnje svestrana i valjano dizajnirana. Sa ovim jezikom možete mnogo da postignete, a posle njega ćete mnogo lakše da učite druge jezike.

Ako vas još uvek zbunjuju pojmovi kao što su program, programski jezik, ili uopšte Java, ne paničite. Sve će početi da se razjašnjava na sledećem času, „Pisanje prvog programa”, u kojem se polako odvija proces pisanja Java programa.

Radionica

Pitanja i odgovori

P. **BASIC? C++? Smalltalk? Java? Šta znače imena ovih jezika?**

O. Ime BASIC je akronim: Beginner's All-purpose Symbolic Instruction Code (univerzalni simbolički kôd instrukcija za početnike). C++ je programski jezik stvoren kao unapređenje jezika C, koji je i sam unapređenje programskog jezika B. Smalltalk je inovativan objektno orijentisani jezik razvijen 1970-ih godina čije su mnoge ideje prihvaćene u Javi. Java je suprotna tradiciji da se za ime jezika uzima akronim ili izraz koji nešto znači. To je prosto ime koje se tvorcima Jave najviše svidelo, nasuprot imenima WebRunner, Silk, Ruby i još nekim. (Programski jezik Ruby tada još nije postojao.) Kada ja budem napravio vlastiti programski jezik, nazvaću ga Salsa. Svi vole salsu.

P. **Zašto su interpretirani jezici sporiji od prevedenih?**

O. Sporiji su iz istog razloga iz kojeg je prevodilac koji interpretira živi govor na stranom jeziku sporiji od prevodioca koji prevodi štampani govor. Prevodilac uživo mora da razmisli o svakoj rečenici kako se dešava, dok drugi prevodilac može da radi na govoru kao celini i koristi prečice da ubrza proces. Prevedeni jezici mogu da budu mnogo brži od interpretiranih zato što rade neke stvari zbog kojih se program efikasnije izvršava.

P. **Da li odgovarate samo na pitanja o Javi?**

O. Nikako. Pitajte bilo šta.

P. **Dobro, koja je najniža ocena ikad dodeljena u TV seriji** *Plesanje sa zvezdama***?**

O. Najgori ples slavnih takmičara u čuvenoj ABC TV emisiji u Sjedinjenim Državama izveo je reper Master P tokom druge sezone 2006. godine. Njegov paso doble sa profesionalnom plesačicom Ešli DelGrosso postigao je najnižu ocenu: 8. Sudije Len Goodman i Bruno Tonioli su mu dodelili po 2, a sudija Carrie Ann Inaba 4.

Sudija Tonioli je komentarisao: „To je bila noćna mora. Izgledali ste kao dete u trgovini u potrazi za svojom majkom."

Goodman je rekao: „Znam da gledaoci misle da su bili ljubazni kad su vas vraćali. Nisu bili. Bili su surovi – prema Ešli, prema sudijama."

Inaba: „Zaista sam smatrala da je to tvoj najbolji ples".

Master P je za nastup trenirao samo 20 sati, u poređenju sa 130 koliko su trenirali ostali takmičari u tom delu sezone. Takođe je odbio da nosi plesne cipele i nastupio je u košarkaškim patikama. Taj ples je bio njegov poslednji, pre nego što je glasanjem otpao.

Ples se može videti na YouTube-u, gde jedan komentator piše: „Svaka čast ako ovo gledate samo da biste videli kako ovo dvoje veslaju".

Za Plesanje sa zvezdama na međunarodnom nivou, TV realiti zvezda Farmer Dave Graham dobio je najmanji mogući rezultat – po 1 od svakog sudije, tj. ukupno 4 – u Australiji 2007. godine. On sam je delio snimak na svom YouTube nalogu, uz opis: „Možda nije najbolji pokušaj za sticanje slave, ali uzmite svoje svetske rekorde gde god možete."

Kviz

Testirajte svoje znanje o materijalu koji je obrađen na ovom času pomoću sledećih pitanja.

1. Šta od sledećeg NIJE razlog da ljudi smatraju kako je programiranje računara strašno teško?
   - **A.** Programeri šire tu famu da bi unapredili svoje izglede na uspeh.
   - **B.** Žargon i akronimi gde god se okreneš.
   - **C.** Oni kojima je programiranje previše teško mogu da dobiju pomoć od vlade.
2. Koja vrsta alatke izvršava računarski program tako što ga proučava red po red?
   - **A.** Spora alatka
   - **B.** Interpreter
   - **C.** Prevodilac
3. Zašto se James Gosling zavukao u svoju kancelariju i napravio Javu?
   - **A.** Nije bio zadovoljan jezikom koji je koristio u jednom projektu.
   - **B.** Njegova rok grupa nije uspevala da dobije nastup.
   - **C.** Kad na poslu ne možete da koristite YouTube, Internet je dosadan.

Odgovori

1. **C.** Ni pisci računarskih knjiga nisu dobili pomoć.
2. **B.** Interpreter tumači red po red. Prevodioci ispitaju instrukcije unapred pa program može brže da se izvrši.
3. **A.** Kočio ga je C++. Tada u 1991-oj kad je napravio Javu, YouTube nije ni postojao.

Aktivnosti

Ako želite bolje da upoznate predmete Java i programiranje računara, uradite sledeće:

- Da biste saznali više o razlozima za učenje Jave, pročitajte ovaj prilog u blogu Oracle University: https://blogs.oracle.com/oracleuniversity/10-reasons-why-you-should-consider-learning-java.
- U govornom jeziku, napišite skup instrukcija za konvertovanje temperature iz Celzijusovih stepeni u Farenhajtove. Podelite instrukcije na što više kratkih redova od po jedne rečenice.

Da biste videli rešenja aktivnosti na kraju svakog časa, posetite veb sajt knjige na adresi www.java24hours.com.