
Početak rada sa jezikom Java za Eclipse

Naučite jezik i razvojnu platformu



Bernhard Steppan

Agencija Eho
www.infoelektronika.net

- Sva prava zadržana. Nijedan deo ove knjige ne sme biti reprodukovan u bilo kom materijalnom obliku, uključujući fotokopiranje ili slučajno ili nenamerno smeštanje na bilo koji elektronski medijum sa ili uz pomoć bilo kog elektronskog sredstva, bez pismenog odobrenja nosioca autorskih prava osim u skladu sa odredbama zakona o autorskim pravima, dizajnu i patentima iz 1988. godine ili pod uslovima izdatim od Copyright Licensing Agency Ltd, 90 Tottenham Court Road, London, England W1P 9HE. Prijave za pismene dozvole radi štampanja bilo kog dela ove publikacije upućuje se izdavaču ove knjige.
- Izjava: Autor i izdavač su uložili najveće napore da bi se obezbedila tačnost informacija sadržanih u ovoj knjizi. Autor i izdavač ne mogu da pretpostave neprijatnosti i ovom izjavom isključuju bilo kakvu odgovornost za bilo koju stranku koja bi imala gubitke ili štetu uzrokovanu greškama ili propustima u ovoj knjizi, bez obzira da li su greške ili propusti nastali usled nemara, nezgode ili bilo kog drugog razloga.

ISBN 978-86-80134-51-2

Početak rada sa jezikom Java za Eclipse

Naslov originala: Getting Started With Java Using Eclipse

Autor: Bernhard Steppan

Prevod: Biljana Tešić

Izdaje i štampa: Agencija Eho, Niš

e-mail: redakcija@infoelektronika.net

Tiraž: 200

Predgovor	21
Poglavlje 1 Osnove programiranja.....	27
1.1 Uvod	27
1.2 Jezik mašinskog sveta	28
1.3 Programski jezici visokog nivoa	30
1.4 Razvojno okruženje	31
1.4.1 Kompajler	31
1.4.2 Uređivač	31
1.4.3 Upravljanje projektom	31
1.5 Okruženje za izvršenje	32
1.6 Rezime	32
1.7 Literatura	33
1.8 Vežbe	33
2 Pregled tehnologije	35
2.1 Uvod	35
2.2 Pregled	36
2.2.1 Počeci Jave	36
2.2.2 Period razvoja Jave	37
2.2.3 Sadašnjost i budućnost Jave	38
2.3 Zašto Java?	39
2.3.1 Lak za čitanje	39
2.3.2 Objektno-orijentisan	39
2.3.3 Bezbedan i robusan	39
2.3.4 Veoma moćan	40
2.3.5 Univerzalno upotrebljiv	40
2.3.6 Besplatan	40
2.3.7 Otvoreni kod	40
2.3.8 Lako prenosiv	40
2.3.9 Lako proširiv	41
2.3.10 Jednostavan za razvoj i testiranje	41
2.4 Šta pripada Javi?	42
2.4.1 Java programski jezik	42
2.4.2 Java virtuelna mašina	42
2.4.3 Java biblioteke klasa	44
2.4.4 Java razvojne alatke	44
2.5 Java verzije	45

2.6 Java izdanja	45
2.6.1 Java Standard Edition	45
2.6.2 Java Enterprise Edition	45
2.6.3 Java Micro Edition	45
2.7 Rezime	46
2.8 Literatura	46
2.9 Vežbe	46
Poglavlje 3 Objektno-orijentisano programiranje	47
3.1 Uvod	47
3.2 Pregled	48
3.3 Objekat	49
3.4 Klasa	50
3.4.1 Svojstva	50
3.4.1.1 Identifikator	51
3.4.2 Metodi	52
3.5 Apstrakcija	54
3.6 Nasleđivanje	55
3.6.1 Osnovne klase	56
3.6.2 Izvedene klase	57
3.6.3 Višestruko nasleđivanje	57
3.7 Zaštita pristupa	58
3.8 Veze	60
3.8.1 Veze bez nasleđivanja	60
3.8.1.1 Asocijacija	61
3.8.1.2 Agregacija	61
3.8.1.3 Kompozicija	62
3.8.2 Veze nasleđivanja	62
3.9 Nedostaci dizajna	64
3.10 Refaktorisanje	65
3.11 Modelovanje	65
3.12 Postojanost	65
3.13 Polimorfizam	65
3.13.1 Statički polimorfizam	66
3.13.2 Dinamički polimorfizam	66
3.14 Pravila dizajna	67
3.15 Rezime	67

3.16 Literatura	68
3.17 Vežbe	68
Poglavlje 4 Razvojno okruženje.....	69
4.1 Uvod	69
4.2 Instalacija	70
4.2.1 Operativni sistem	70
4.2.2 Instaliranje Jave	71
4.2.3 Instaliranje Eclipse-a	74
4.2.4 Instaliranje primera programa	79
4.2.5 Provera instalacije	82
4.3 Uvod u Eclipse	84
4.3.1 Pregled	84
4.3.2 Workbench	84
4.3.3 Perspektive, prikazi i uređivači	85
4.3.4 Package Explorer	86
4.3.5 Java uređivač	87
4.3.6 Formater koda	90
4.3.7 Build System	92
4.3.8 Debager	92
4.3.9 Modularna struktura	93
4.3.10 Eclipse radni prostor	94
4.3.11 Ažuriranje softvera	96
4.3.12 Sistem za pomoć	97
4.4 Rezime	98
4.5 Literatura	98
4.6 Vežbe	99
Poglavlje 5 Struktura programa	105
5.1 Uvod	105
5.2 Pregled	106
5.3 Jezički elementi	107
5.3.1 Komentari	108
5.3.2 Paketi	108
5.3.3 Klase	108
5.3.4 Metodi	109
5.3.5 Iskazi	110
5.4 Struktura programa	112

5.5 Tok programa	113
5.6 Rezervisane ključne reči	114
5.7 Rezime	115
5.8 Uputstvo	116
5.9 Vežbe	116
Poglavlje 6 Promenljive	117
6.1 Uvod	117
6.2 Pregled	118
6.2.1 Svrha promenljive	118
6.2.2 Tipovi promenljivih.....	118
6.2.3 Upotreba promenljivih	119
6.3 Lokalne promenljive	121
6.4 Parametri.....	122
6.5 Promenljive instance.....	123
6.5.1 Pojedinačne promenljive instance.....	123
6.5.2 Promenljiva instance „this“	124
6.6 Promenljive klase.....	126
6.7 Konstante	128
6.8 Rezime	130
6.9 Literatura.....	131
6.10 Uputstvo	131
6.11 Vežbe	131
Poglavlje 7 Iskazi	133
7.1 Uvod	133
7.2 Pregled	134
7.2.1 Svrha iskaza	134
7.2.2 Tipovi iskaza	135
7.3 Deklaracija	135
7.4 Dodela	137
7.4.1 Struktura Java dodele.....	137
7.4.2 Java zadaci nisu jednaki matematičkim jednačinama	137
7.4.3 Da li je $x = y$ jednako $y = x$?	138
7.4.4 Kombinacija deklaracije i dodele vrednosti	139
7.5 Blok	140
7.6 Poziv promenljive	143

7.7 Poziv metoda	144
7.8 Rezime	145
7.9 Literatura	146
7.10 Uputstvo	146
7.11 Vežbe	146
Poglavlje 8 Osnovni tipovi podataka	147
8.1 Uvod	147
8.2 Pregled	148
8.2.1 Svrha osnovnih tipova podataka	148
8.2.2 Tipovi osnovnih tipova podataka	148
8.2.3 Upotreba osnovnih tipova podataka	148
8.3 Celobrojni tipovi podataka	152
8.3.1 Tip podataka „byte“	152
8.3.2 Tip podataka „short“	153
8.3.3 Tip podataka „int“	154
8.3.4 Tip podataka »dugo«	155
8.4 Tipovi podataka sa pokretnom tačkom.....	155
8.4.1 Tip podataka „float“	156
8.4.2 Tip podataka „double“	157
8.5 Tip podataka karaktera.....	158
8.6 Bulov tip podataka.....	158
8.7 Rezime	159
8.8 Literatura	161
8.9 Uputstvo	161
8.10 Vežbe	161
Poglavlje 9 Klase i objekti	163
9.1 Uvod	163
9.2 Pregled	164
9.2.1 Svrha klase	164
9.2.2 Tipovi klasa.....	165
9.2.3 Definicija klasa	166
9.2.4 Upotreba klasa	166
9.3 Anonimne klase	169
9.3.1 Definicija konkretnih klasa	169
9.3.2 Kreiranje objekata konkretne klase	170

9.3.3 Unutrašnje klase.....	171
9.3.4 Lokalne klase.....	173
9.3.5 Anonimne klase	174
9.3.6 Nasleđivanje.....	176
9.4 Apstraktne klase	181
9.5 Interfejsi	182
9.6 Generički tipovi	186
9.6.1 Definicija generičkih klasa.....	186
9.6.2 Kreiranje generičkih objekata.....	187
9.7 Rezime	191
9.8 Literatura	192
9.9 Uputstvo	192
9.10 Vežbe	192
Poglavlje 10 Nabranja.....	193
10.1 Uvod	193
10.2 Pregled	193
10.2.1 Svrha enuma	193
10.2.2 Definicija i deklaracija enuma	195
10.2.3 Upotreba enuma.....	196
10.3 Enum klase	197
10.3.1 Konstruktor	198
10.3.2 Metod „value()“	198
10.3.3 Zasebna klasa Simple Enum	198
10.3.4 Zasebna proširena klasa Enum	200
10.3.5 Unutrašnja proširena klasa Enum	201
10.4 Rezime.....	202
10.5 Literatura	202
10.6 Uputstvo	203
10.7 Vežbe	203
Poglavlje 11 Nizovi	205
11.1 Uvod	205
11.2 Pregled	206
11.2.1 Svrha nizova	206
11.2.2 Tipovi nizova.....	206
11.2.3 Upotreba nizova	207

11.3 Rezime	211
11.4 Uputstvo	212
11.5 Vežbe	212
Poglavlje 12 Metodi.....	213
12.1 Uvod	213
12.2 Pregled	214
12.2.1 Svrha metoda	214
12.2.2 Tipovi metoda	215
12.2.3 Definicija metoda	216
12.2.4 Upotreba metoda	219
12.3 Konstruktori	221
12.3.1 Podrazumevani konstruktori	222
12.3.2 Konstruktori bez parametara.....	223
12.3.3 Konstruktori sa parametrima.....	224
12.4 Destruktori.....	226
12.5 Operacije.....	228
12.6 Metodi getter.....	229
12.6.1 Definicija	229
12.6.2 Upotreba	231
12.7 Metodi setter.....	232
12.7.1 Definicija	232
12.7.2 Upotreba	233
12.8 Rezime	234
12.9 Literatura	235
12.10 Uputstvo	235
12.11 Vežbe	235
Poglavlje 13 Operatori	237
13.1 Uvod	237
13.2 Pregled	238
13.2.1 Tipovi operatora	238
13.2.2 Prioritet operatora.....	238
13.3 Aritmetički operatori.....	239
13.3.1 Unarni plus operator	239
13.3.2 Unarni minus operator	240
13.3.3 Operator sabiranja.....	241

13.3.4 Operator oduzimanja	242
13.3.5 Operator množenja.....	243
13.3.6 Operator deljenja	243
13.3.7 Operator ostatka deljenja	244
13.3.8 Operator za prefiksno inkrementiranje	244
13.4 Relacioni operatori.....	247
13.4.1 Operator „jednako“	248
13.4.2 Operator „nije jednako“	248
13.4.3 Operator „manje od“	249
13.4.4 Operator „manje od ili jednako“	249
13.4.5 Operator „veće od“	250
13.4.6 Operator „veće ili jednako“	250
13.4.7 Operator poređenja tipova	251
13.5 Logički operatori	254
13.5.1 Operator logičkog komplementa	254
13.5.2 Operator AND	254
13.5.3 Operator OR.....	256
13.5.4 Ternarni operator.....	256
13.6 Operatori nad bitovima	257
13.7 Operatori dodele	257
13.8 Operator new	258
13.9 Operater konverzija tipa	259
13.10 Operatori pristupa.....	261
13.10.1 Operator tačka.....	261
13.10.2 Lambda operator.....	263
13.11 Rezime.....	264
13.12 Literatura	264
13.13 Uputstvo	264
13.14 Vežbe	264
Poglavlje 14 Uslovni iskazi	265
14.1 Uvod	265
14.2 Pregled	266
14.3 Iskazi „If then else“	266
14.4 Ternarni operator	267
14.5 Iskaz switch	269
14.5.1 Iskaz switch na nivou Jave 6	269

14.5.2 Iskaz switch na nivou Jave 7	270
14.5.3 Iskaz yield	271
14.5.4 Lambda izraz	272
14.6 Rezime	273
14.7 Literatura	273
14.8 Uputstvo	273
14.9 Vežbe	273
Poglavlje 15 Petlje	275
15.1 Uvod	275
15.2 Pregled	276
15.2.1 Svrha petlji	276
15.2.2 Tipovi petlji	276
15.3 Petlja while	277
15.4 Petlja do	278
15.5 Jednostavna petlja for	279
15.6 Proširena petlja for	280
15.7 Rezime	281
15.8 Literatura	282
13.13 Uputstvo	282
13.14 Vežbe	282
Poglavlje 16 Paketi i moduli	283
16.1 Uvod	283
16.2 Pregled	284
16.3 Paketi	284
16.3.1 Uvoz klasa	284
16.4 Moduli	288
16.5 Rezime	291
16.6 Uputstvo	291
16.7 Vežbe	291
17.1 Uvod	293
17.2 Pregled	294
17.2.1 Motivacija	294
17.2.2 Tipovi grešaka	294
17.2.3 Korišćenje rukovanja izuzecima	294
17.3 Osnovna klasa „Throwable“	297

17.4 Klasa „Error“	298
17.4.1 Potklasa „OutOfMemoryError“	299
17.4.2 Potklasa „StackOverflowError“	301
17.5 Klasa „Exception“	302
17.5.1 Potklasa „RuntimeException“	302
17.5.2 Potklasa „IOException“	302
17.5.3 Samoprogramirani izuzeci.....	303
17.6 Rezime	306
17.7 Literatura	307
17.8 Uputstvo	307
17.9 Vežbe	307
Poglavlje 18 Dokumentacija.....	309
18.1 Uvod	309
18.2 Pregled	310
18.3 Komentari linije	310
18.4 Blok komentari	311
18.5 Komentari o dokumentaciji.....	311
18.6 Rezime.....	312
18.7 Literatura	314
18.8 Uputstvo	314
18.9 Vežbe	314
Poglavlje 19 Anotacije	315
19.1 Uvod	315
19.2 Pregled	316
19.2.1 Svrhe anotacija.....	316
19.2.2 Tipovi anotacija	316
19.2.3 Unapred definisane anotacije.....	317
19.2.4 Upotreba anotacije.....	318
19.3 Anotacije za kontrolu kompajlera.....	320
19.3.1 Anotacije „Deprecated“	320
19.3.2 Anotacije „SuppressWarnings“	322
19.3.3 Anotacija „Override“	328
19.4 Rezime	331
19.5 Literatura	331
19.6 Uputstvo	331

19.7 Vežbe	331
Poglavlje 20 Razvojni procesi	335
20.1 Uvod	335
20.2 Pregled	336
20.2.2 Aktivnosti	338
20.2.3 Alatkke	339
20.3 Faza planiranja.....	339
20.3.1 Pojašnjenje porudžbine.....	339
20.3.2 Snimanje zahteva.....	339
20.4 Faza konstrukcije.....	340
20.4.1 Analiza	340
20.4.2 Dizajn	341
20.4.3 Implementacija	342
20.4.4 Test.....	354
20.5 Faza rada.....	359
20.5.1 Raspoređivanje.....	359
20.5.2 Održavanje	361
20.6 Rezime	361
20.7 Literatura	362
20.8 Vežbe	362
Poglavlje 21 Okruženje za izvršenje	363
21.1 Uvod	363
21.2 Pregled	363
21.3 Bajt-kod.....	364
21.4 Java virtuelna mašina	368
21.4.1 Veštački računar	368
21.4.2 Režim interpretatora	369
21.4.3 Režim JIT kompajlera	370
21.4.4 Režim Hotspot.....	370
21.4.5 Sakupljač smeća.....	371
21.5 Biblioteke.....	371
21.5.1 Izvorne biblioteke.....	371
21.5.2 Biblioteke klasa	372
21.5.3 Resursi i datoteke sa svojstvima	372
21.6 Prenosivost	373

21.6.1 Binarno kompatibilan bajt-kod	373
21.6.2 Preduslovi za prenos.....	373
21.7 Pokretanje programa	374
21.7.1 Pokretanje skripta	374
21.7.2 Izvorni omotač	375
21.8 Konfiguracija JVM-a.....	377
21.9 Rezime.....	377
21.10 Literatura.....	379
21.11 Vežbe	379
Poglavlje 22 Biblioteka klasa	381
22.1 Uvod	381
22.2 Pregled	382
22.2.1 Oblasti primene.....	383
22.2.2 Ponovna upotreba.....	383
22.2.3 Dokumentacija	383
22.2.4 Proširenje jezika	383
22.3 Java Standard Edition.....	383
22.3.1 Osnovne klase.....	384
22.3.2 Klasa „System“	392
22.3.3 Programske niti	395
22.3.4 Tokovi (Streams).....	396
22.3.5 Svojstva.....	398
22.3.6 Kontejnerske klase	400
22.3.7 Apstraktne alatke za prozore	401
22.3.8 Swing.....	410
22.3.9 JavaBeans	413
22.3.10 Apleti	414
22.3.11 Java Database Connectivity (JDBC).....	414
22.3.12 Java Native Interface	415
22.3.13 Remote Method Invocation.....	416
22.4 Java Enterprise Edition	417
22.4.1 Entity Beans	418
22.4.2 Session Beans.....	418
22.4.3 Message Driven Beans.....	418
22.4.4 Interfejsi.....	418
22.5 Java Micro Edition	419

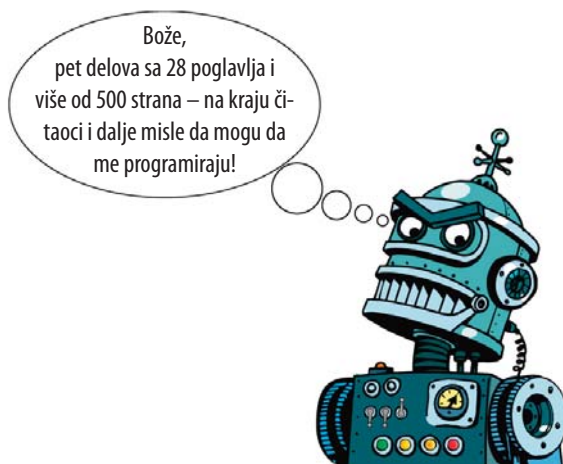
22.6	Biblioteke eksternih klasa	420
22.6.1	Apache Software Foundation	420
22.6.2	Eclipse zajednica	420
22.6.3	SourceForge	420
22.6.4	Drugi softver otvorenog koda	420
22.6.5	Komercijalni softver	421
22.7	Rezime	421
22.8	Literatura	422
22.9	Vežbe	422
	Poglavlje 23 Pravila	423
23.1	Uvod	423
23.2	Pregled	424
23.3	Konvencije pisanja	425
23.4	Zaštita pristupa	425
23.4.1	Četiri nivoa zaštite pristupa	425
23.4.2	Nivo pristupa „private“	426
23.4.3	Nivo pristupa „default“	426
23.4.4	Nivo pristupa „protected“	426
23.4.5	Nivo pristupa „public“	427
23.4.6	Studija slučaja	427
23.4.7	Opseg promenljivih	433
23.5	Redosled razvoja	437
23.5.1	Tačka ispred crtice	438
23.6	Konverzija tipa	441
23.6.1	Implicitna konverzija	441
23.6.2	Eksplcitna konverzija	443
23.7	Polimorfizam	446
23.7.1	Preklapanje metoda	446
23.7.2	Prepisivanje metoda	448
23.8	Rezime	452
23.9	Literatura	453
23.10	Vežbe	453
	Poglavlje 24 Algoritmi	455
24.1	Uvod	455
24.2	Pregled	456

24.2.1	Razvoj algoritama	456
24.2.2	Tipovi algoritama	457
24.2.3	Upotreba algoritama	457
24.3	Razvijanje algoritama	457
24.3.1	Algoritmi za sortiranje.....	457
24.3.2	Grafički algoritmi.....	458
24.4	Upotreba algoritma	466
24.4.1	Algoritmi za sortiranje.....	466
24.4.2	Algoritmi pretraživanja.....	468
24.5	Rezime	470
24.6	Literatura	470
24.7	Vežbe	470
Poglavlje 25	Swing programi.....	473
25.1	Uvod.....	473
25.2	Zahtevi	474
25.3	Analiza i dizajniranje	474
25.3.1	Korisnički interfejs.....	475
25.3.2	Programska logika	475
25.4	Implementacija.....	478
25.4.1	Pokrenite Eclipse pomoću radnog prostora „Exercises“	478
25.4.2	Kreirajte novi Java projekat „Swing Programs“	479
25.4.3	Kreiranje nove klase „CourseStatisticsApp“	479
25.4.4	Implementacija klase „CourseStatisticsApp“	479
25.4.5	Kreiranje nove klase „MainWindow“	480
25.4.6	Implementacija klase „MainWindow“	481
25.4.7	Implementacija klase „CsvParser“	492
25.4.8	Implementacija klase „TableFilter“	493
25.5	Test	496
25.6	Raspoređivanje	496
25.7	Rezime.....	498
Poglavlje 26	Česte greške	501
26.1	Uvod	501
26.2	Java greške.....	501
26.2.1	Nije moguće kreirati statičku referencu na nestatičko polje.....	501
26.2.2	Izlaz „null“ vrednosti	502

26.2.3	NullPointerException	503
26.2.4	Nedostajanje prekida u iskazu case	505
26.2.5	Netačno poređenje	506
26.2.6	Neobrađeni izuzeci	509
26.2.7	NoClassDefFoundError	510
26.2.8	ClassNotFoundException	510
26.3	Eclipse greške.....	510
26.3.1	Eclipse nije mogao da se pokrene	510
26.3.2	Haotična Eclipse perspektiva	510
26.3.3	Nedostajući prozor	511
26.4	Rezime	511
26.5	Literatura	511
27	Rečnik	513

Predgovor

Java je trenutno neosporno najvažniji programski jezik. Mnogi ljudi bi želeli da nauče Javu, ali početak rada nije lak pošto programiranje pomoću Jave zahteva najmanje dve stvari: učenje programskog jezika i razvojnog okruženja. Pomoću mnogih primera u ovoj knjizi je pokazano kako je jezik strukturiran. Osim toga, u knjizi je korišćeno razvojno okruženje Eclipse kao primer moćne alatke za podučavanje razvoja Java programa.



Kroz knjigu vas prati Robert iz mašinskog sveta.

U prvom delu knjige „**Osnove**“, stičete Java i Eclipse osnovno znanje. U tom delu je postavljen temelj programiranja, obezbeđen je kratak pregled Java tehnologije i pokazano šta je posebno u objektno-orijentisanom programiranju. Taj deo se završava poglavljem o razvojnom okruženju Eclipse.

U drugom delu koji se zove „**Java jezik**“, sve se svodi na suptilnosti Java jezika i tu nastaju prve male Java aplikacije pomoću finog spoja znanja i praktičnih vežbi. Taj deo obezbeđuje kombinaciju dela znanja i praktičnih vežbi. Na kraju svakog poglavlja ćete pronaći zadatke koje možete sami da rešite. Uspех u učenju ćete proveriti u rešenjima zadataka na kraju ove knjige.

Java tehnologija je fokus trećeg dela „Java tehnologija“ koji vas takođe upozna sa pravilima kojih bi trebalo da se pridržavate tokom programiranja i šta su biblioteke klasa i koje su njihove prednosti. Osim toga, naučićete kako da testirate programe i šta su algoritmi i kako da ih programirate.

Veći Java projekat je fokus četvrtog dela. Ovde ćete da primenite sve prethodne elemente u aplikaciji pomoću grafičkog korisničkog interfejsa. Projekat pokazuje kako da se razvije veća aplikacija „deo po deo“ pomoću razvojnog okruženja Eclipse.

Peti deo „Dodatak“ završava ovu knjigu rešenjima zadataka, sa osnovama obrade informacija i poglavljem o najčešćim greškama koje se mogu javiti kada koristite Eclipse i Glossary.

Radnja

Kao radnju, knjigu sam zasnovao na (izmišljenom) kursu programiranja „Java za Eclipse“ koji je profesor Rot predavao za četiri studenta. Kurs programiranja u knjizi prati robot koji se zove „Robert“ i, između ostalih, uglavnom ovih pet likova u knjizi:



Kurs programiranja sa Lukasom, Anom, profesorom Rotom, Julijom i Florijanom

Kome je namenjena ova knjiga?

Ova knjiga je namenjena aktivnim čitaocima. Ne želite gotova rešenja, želite da sami programirate. Ako ne budete aktivno programirali i ne budete spremni dok se ne pokrene program koji ste sami napisali, nećete naučiti Javu. Knjiga sadrži primamljiv broj spremnih programiranih primera koje možete pokrenuti klikom na jedno dugme. Posegnite za primerima rešenja samo ako dođe do problema. Prvo pokušajte sami da pristupite programima i da učite iz grešaka. Samo aktivnim programiranjem ćete naučiti Javu i Eclipse razvojno okruženje.

Dodatni materijal

Knjiga sadrži mnoštvo primera koji se lako mogu uvesti u Eclipse okruženje kao rešenja. Možete ih lako preuzeti sa početne stranice izdavačke kuće „Elektor“ www.elektor.com/books/programming. Među tim preuzimanjima naći ćete i dodatno poglavlje, koje nije odštampano zbog ograničenja prostora. U njemu je objašnjeno programiranje takozvanih terminalskih programa.

Konvencije o fontovima

Različiti delovi teksta su istaknuti na sledeći način radi bolje čitljivosti:

Tekstualni deo	Značenje
Tipovi podataka u tekstu	<i>Person</i>
Tipovi podataka u naslovima	„Person“
Ključne reči u tekstu	<i>implementira</i>
Ključne reči u naslovima	„implementira“
Promenljive u tekstu	roland
Promenljive u naslovima	„roland“
Prozor (grafički korisnički interfejs)	Eclipse IDE Launcher
GUI element (grafički korisnički interfejs)	Finish
Meni (grafički korisnički interfejs)	File
Komanda menija (grafički korisnički interfejs)	Menu → File → New → Java → Project
Datoteke	<i>Samples.zip</i>
Putanje direktorijuma	<i>C:/Programs/Eclipse</i>
Listinzi (izvorni kod iz primera programa)	<pre>1 package programmingcourse; 2 public class Robot { 3 (...) 4 }</pre>
Izlaz programa	Result = true
URL	http://eclipse.org
(...)	Zbog ograničenja prostora, deo izvornog koda nedostaje

Priznanja

Želeo bih da se zahvalim svima koji su me podržali u pisanju ove knjige: izdavačkoj kući „Elektor“ i mom uredniku Ferdinandu te Valvartu na poverenju u moj rad i njegovom velikom strpljenju. Želeo bih da se zahvalim izdavačkoj kući „Hanser“ koja je dala dozvolu za prevod originalnog nemačkog rukopisa.

Kao i uvek, moja supruga Kristijana pružila je veliku podršku ovom projektu. Hvala mnogo na pomoći! Takođe bih želeo da se zahvalim Alini Neasu, koja je ispravila pravopisne greške iz mog rukopisa knjige. Veliko hvala i Valeriju Kačaevu (Studiostoks), od koga potiču šabloni crtanih likova robota.

Kako da stupite u kontakt sa nama

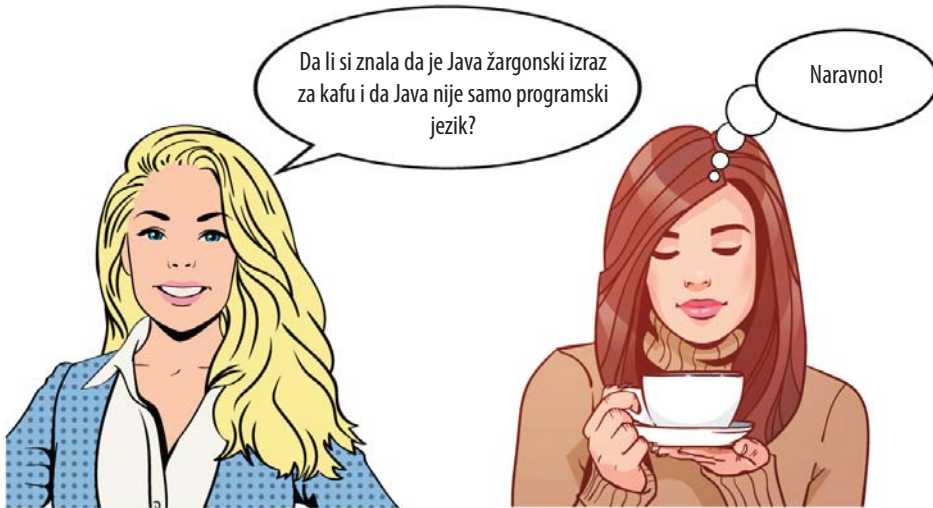
Uprkos najvećoj pažnji, nije uvek moguće izbeći previd nekih grešaka u knjizi. Ako pronađete greške, imate predloge za poboljšanje ili pitanja, samo mi pošaljite imejl na adresu java-eclipse@steppan.net. Odgovoriću na vaša pitanja u najkraćem mogućem roku i pokušati da uvrstim vaše predloge za poboljšanje u naredna izdanja. Najnovije dodatke i dodatne informacije možete pronaći na adresi <http://www.programmingcourse.net>. Sada se nadam da ćete uživati u čitanju i razvoju svojih Java programa koristeći Eclipse!

Bernhard Steppan

DEO I

Osnove

Da biste mogli da razvijate računarske programe, potrebno je da naučite osnove. Poglavlje „Osnove programiranja“ postavlja temelj za programiranje Java aplikacija. U poglavlju 2, „Pregled tehnologije“, naučićete šta Java ima zajedničko sa drugim programskim jezicima i po čemu se Java razlikuje od drugih jezika.



Slika 1: Da biste razvili računarske programe, potrebno je da naučite osnove

Zatim se knjiga nastavlja poglavljem „Objektno-orijentisano programiranje“. Pokazuje šta je posebno u objektno-orijentisanom programiranju i kako su objektno-orijentisani programi strukturirani. Poglavlje „Razvojno okruženje“ zaključuje taj deo knjige instaliranjem razvojnog okruženja i uvodom u „Eclipse“.

1 Osnove programiranja

1.1 Uvod

Programiranje podrazumeva pisanje računarskih programa. Računarski programi se sastoje od jedne ili više komandi u programskom jeziku. Robot koji se zove Robert predstavlja jednostavan Java program studentima na kursu programiranja profesora Rota (slika 1.1).

```
class Hello {  
    public static void main(String[] args) {  
        System.out.print("Hello!");  
    }  
}
```



Slika 1.1: Robert iz mašinskog sveta je *stručnjak* za mašinske programe.

Studenti kursa programiranja smatraju da postoji mnogo instrukcija za toliko jednostavan program. Ana bi želela da sazna od profesora Rota da li to može biti jednostavnije:



Slika 1.2: Da li je zaista i danas prikladno kucati programe?

„Čak su i Alexa, Cortana i Siri“, kaže profesor Rot, „samo računarski programi.“ Ti programi su razvijeni tako da ljudi mogu da kontrolišu računare pomoću govora. Ali, Alexa & Co. mogu da urade samo nekoliko zadataka za koje su posebno programirani. Ako želite da računar izvršava i druge zadatke, kao što je obrada teksta, morate napisati poseban program za njega. Na primer, ti programi se mogu razviti u Javi.

1.2 Jezik mašinskog sveta

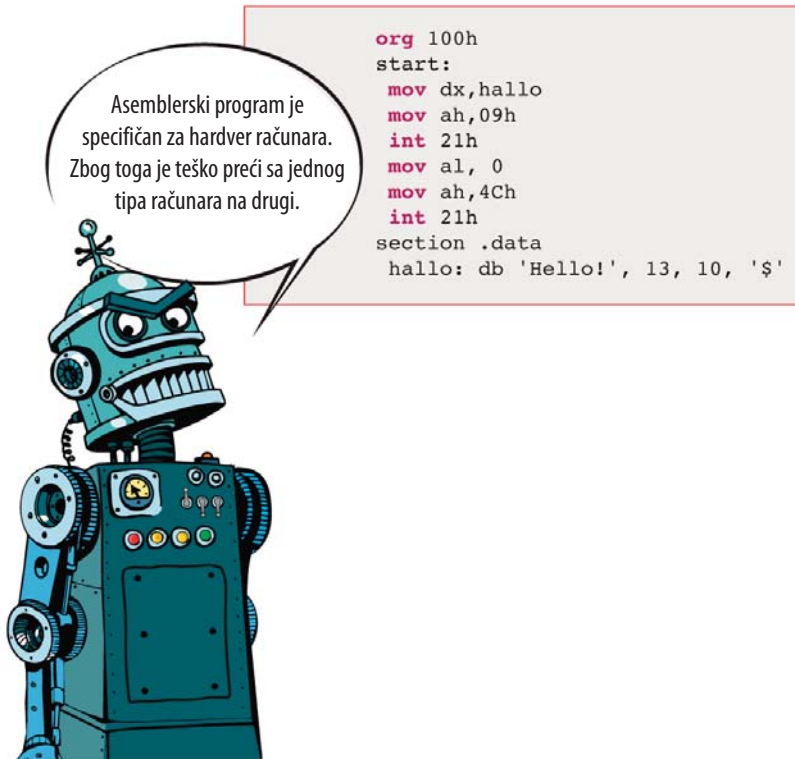
Kada govorimo o računarima danas, uvek mislimo na *digitalne* računare. Te mašine razumeju samo svoj digitalni mašinski jezik. Digitalno označava da računar koristi binarni kod za sve informacije. Stoga se mašinski jezik računara sastoji samo od niza nula i jedinica.

Međutim, mašinski jezik računara, sa njegovim nizovima nula i jedinica, je izuzetno težak za ljudsko razumevanje. Stoga bi bilo potpuno apsurdno programirati računare direktno u mašinskom kodu. To bi potrajalo veoma dugo i verovatnoća grešaka bi bila velika.

Ako želite da programirate računar kao mašina, koristite pomoćni jezik. Taj pomoćni jezik se naziva *asembler*ski jezik ili skraćeno *assembler*. Profesor Rot predstavlja jednostavan primer na svom času programiranja. Baš kao prethodni Java program, on jednostavno ispisuje niz znakova „Hello!“ na ekranu (slika 1.3).

Kurs programiranja profesora Rota smatra da je program *asembler*skog jezika prilično težak za razumevanje. Kako su samo programeri mogli da nauče toliko strašan jezik?

Odgovor je jednostavan, jer u prvim danima računara nije bilo jezika visokog nivoa kao što je Java. Programeri su morali da obrate posebnu pažnju na procesore računara kada su programirali mašinu na asemblerskom jeziku.



Asemblerski program je specifičan za hardver računara. Zbog toga je teško preći sa jednog tipa računara na drugi.

```
org 100h
start:
mov dx,hallo
mov ah,09h
int 21h
mov al, 0
mov ah,4Ch
int 21h
section .data
hallo: db 'Hello!', 13, 10, '$'
```

Slika 1.3: Ovaj asemblerski program takođe ispisuje „Hello!“.

Asemblerski programi su mnogo duži u odnosu na funkcionalno ekvivalentne Java programe. Oni se sastoje od mnogih komandi malih delova koji mogu da urade malo toga ako se koriste samostalno. Zbog toga je potrebno mnogo tih komandi za pisanje većeg programa. Taj program je napisan posebno za jedan tip računara. Teško ga je preneti na drugi tip računara.

Osim visokih troškova razvoja takvih programa, glavni nedostatak asemblerskog jezika je da se teško prenosi sa jednog tipa računara na drugi. Međutim, instrukcije za male delove nemaju samo nedostatke. Prednost im je da ih dobar programer može koristiti za kreiranje veoma jednostavnih i brzih mašinskih programa. Takođe često zahtevaju mnogo manje radne memorije od uporedivih programa napisanih na jeziku visokog nivoa.

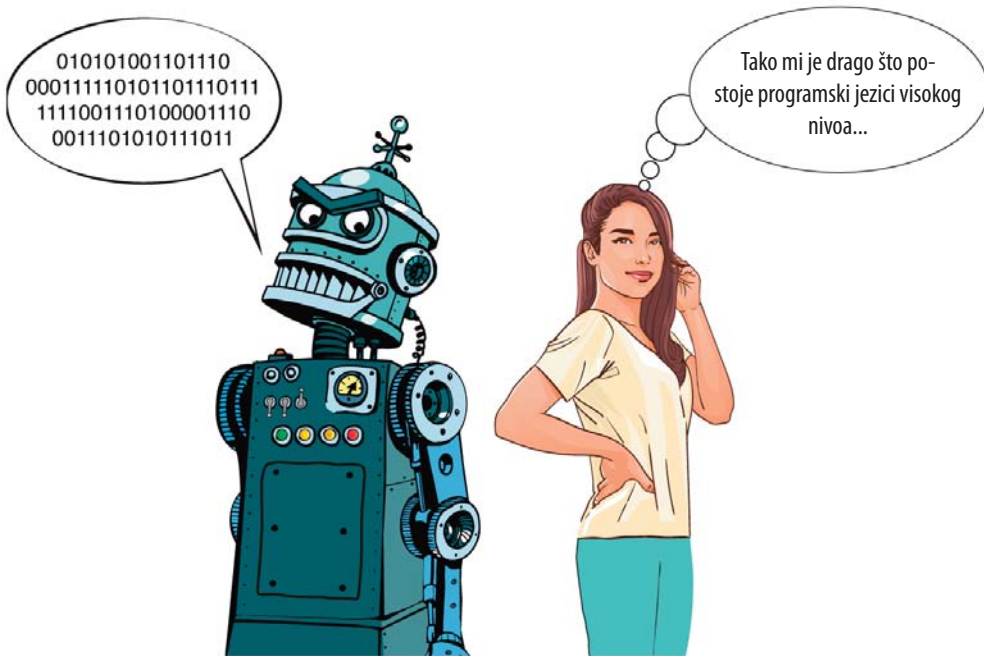
1.3 Programski jezici visokog nivoa

Čini se da je nekako začarano: računari razumeju samo svoj specifični mašinski jezik. Mi, s druge strane, bez posebne obuke za programiranje razumemo samo svoj maternji

jezik i možda jedan ili dva strana jezika. Kako možemo premostiti taj ogroman jaz između mašinskog i ljudskog sveta?

Možemo da razvijemo još moćnije programe kao što su Alexa, Cortana i Siri, tako da računari izvršavaju sve što želimo. Ili možemo da naučimo jezik računara ako želimo da razvijemo specijalne programe za zadatke koje Alexa & Co. ne mogu da urade – ne, to nisu jedine opcije, jer, na sreću, postoji i treći način.

Programiranje složenog programa u assembleru više nije aktuelno. Zbog toga su ljudi vrlo rano počeli da razvijaju programske jezike poput Java jezika. Ti jezici predstavljaju „most“ između (za većinu ljudi) teško razumljivog mašinskog jezika i (za većinu mašina) teško razumljivog ljudskog jezika. Ti jezici se nazivaju programski jezici visokog nivoa ili skraćeno jezici visokog nivoa.



Slika 1.4: Programski jezici visokog nivoa su posrednici između ljudi i mašina.

Čoveku je mnogo lakše da nauči i razume programski jezik visokog nivoa nego jezik mašinskog sveta. Ali, kako to funkcioniše? Kako prevesti jezik visokog nivoa na jezik mašinskog sveta? U tu svrhu osmišljen je trik. Taj trik je specijalan program koji prevodi izvorni kod jezika visokog nivoa kao što je Java u jezik mašinskog sveta. Taj program se naziva kompajler i deo je razvojnog okruženja.

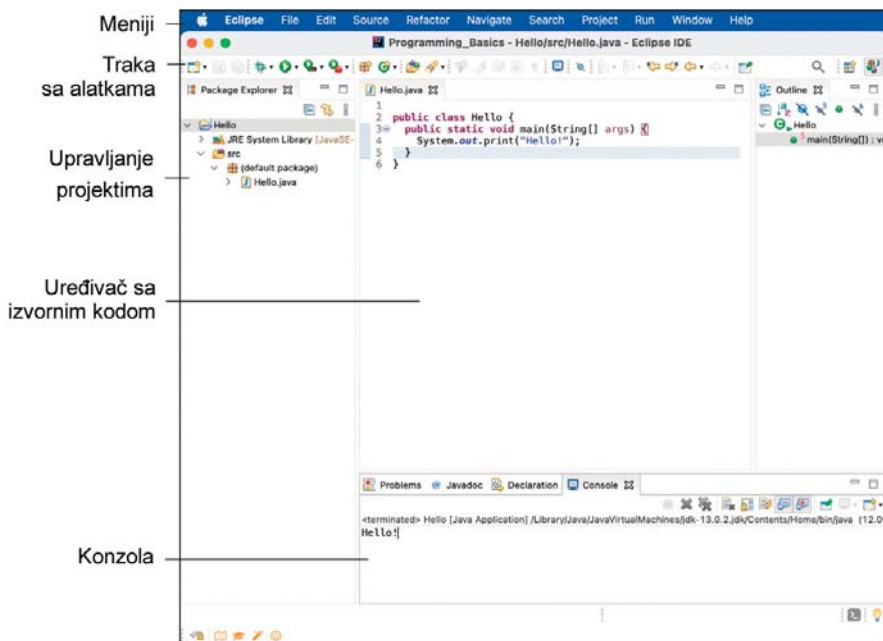
1.4 Razvojno okruženje

1.4.1 Kompajler

Kompajler je jedna od ključnih komponenti razvojnog okruženja kao što je Eclipse. On prevodi izvorni kod Java programa na jezik mašinskog sveta. Izvorni kod je tekst koji se vidi na slikama 1.1 i 1.3.

1.4.2 Uređivač

U uređivač unosite izvorni kod programa kao u program za obradu teksta. Uređivač, takođe, obezbeđuje podršku za razvoj programa, kao što je savet o tome kako da popravite greške koje se prikazuju.



Slika 1.5: Uređivač, kompajler i upravljanje projektom Eclipse razvojnog okruženja.

1.4.3 Upravljanje projektom

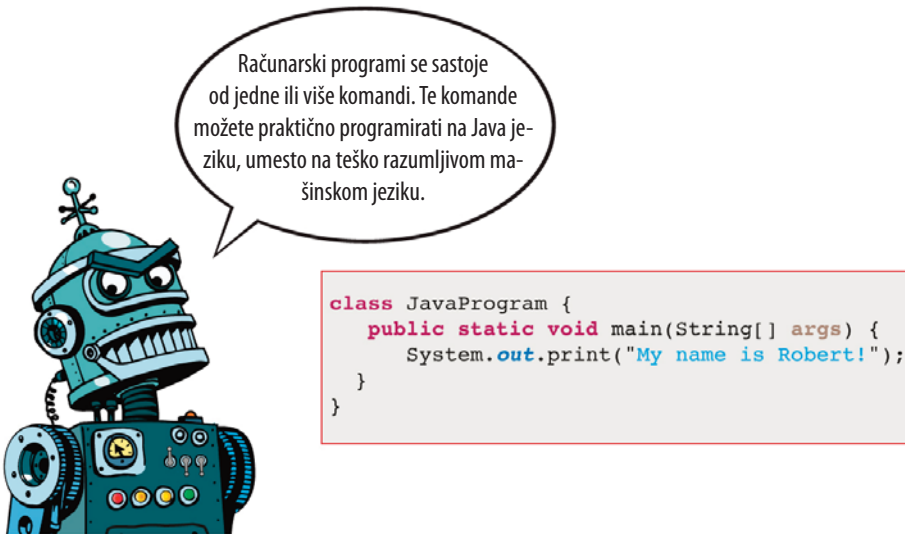
Java programi se obično sastoje od velikog broja datoteka. Da biste mogli da koristite pregled, Java razvojno okruženje ima upravljanje projektima. Pokazuje koje datoteke pripadaju projektu.

1.5 Okruženje za izvršenje

Java programi zahtevaju posebno okruženje za izvršenje. Drugim rečima: Java programi se pokreću samo pomoću dodatnog programa na vašem računaru. To u početku ne primećujete, jer razvojno okruženje Eclipse poziva to okruženje za izvršenje u pozadini kada pokrenete Java program. Da biste razumeli šta je to okruženje za izvršenje, pređite na sledeće poglavlje. U njemu je pokazano kako se Java razvijala i zašto je Java okruženje za izvršenje uopšte neophodno.

1.6 Rezime

Programiranje podrazumeva pisanje računarskih programa. Računarski programi se sastoje od jedne ili više komandi u programskom jeziku. Te komande su napisane u obliku teksta. U programiranju se taj tekst naziva izvorni kod. Računari očekuju komande na mašinskom jeziku. Mi, pak, govorimo svojim ljudskim jezikom. Da bi premostili taj jaz, inženjerski su razvili jezike visokog nivoa. Java je jedan od tih jezika visokog nivoa.



Slika 1.6: Računarski programi se sastoje od komandi programskog jezika.

Da biste preveli program jezika visokog nivoa na mašinski jezik, tako da računar može da ga izvrši, potreban vam je dodatni program. Taj program za prevođenje naziva se kompajler. Kompajler je deo razvojnog okruženja. On se sastoji (između ostalog) još uvek od uređivača i upravljanja projektom. Pomoću uređivača pišete izvorni kod programa. Administracija projekta upravlja različitim datotekama koje pripadaju projektu.