

knihovna programátora

- Podrobný výklad vlastností jazyka od naprostých základů až po pokročilé, běžně neprobírané konstrukce
- Vysvětluje nejenom jak probírané konstrukce používat, ale také proč jsou právě takové
- Využívá zabudované REPL prostředí pro demonstraci vykládaných konstrukcí bez zbytečného pomocného kódu
- Může sloužit současně jako učebnice i referenční příručka
- Ukazuje, jak efektivně experimentovat a využitím prostředí JShell získat okamžité odpovědi



RUDOLF PECINOVSKÝ

Java 9

Kompletní příručka jazyka



knihovna programátora

RUDOLF PECINOVSKÝ

Java 9

Kompletní příručka jazyka

GRADA
Publishing

Upozornění pro čtenáře a uživatele této knihy

Všechna práva vyhrazena. Žádná část této tištěné či elektronické knihy nesmí být reprodukována a šířena v papírové, elektronické či jiné podobě bez předchozího písemného souhlasu nakladatele.

Neoprávněné užití této knihy bude **trestně stíháno**.

Rudolf Pecinovský

Java 9

Kompletní příručka jazyka

Vydala Grada Publishing, a.s.

U Průhonu 22, Praha 7

obchod@grada.cz, www.grada.cz

tel.: +420 234 264 401, fax: +420 234 264 400

jako svou 6768. publikaci

Odpovědný redaktor: Jaroslava Palasová

Návrh vnitřního layoutu: Rudolf Pecinovský

Zlom: Rudolf Pecinovský

Počet stran 560

První vydání, Praha 2018

Vytiskly Tiskárny Havlíčkův Brod, a. s.

© Grada Publishing, a.s., 2018

Cover Design © Grada Publishing, a. s., 2018

Cover Photo © Depositphotos

Názvy produktů, firem apod. použité v knize mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

ISBN 978-80-271-0930-2 (pdf)

ISBN 978-80-271-0715-5 (print)

Všem, kteří se chtějí něco naučit

Stručný obsah

Stručný obsah	6
Podrobný obsah	8
Seznam výpisů programů	22
Seznam obrázků	27
Seznam odboček – podšeděných bloků	29
Úvod	30
Část I: Neobjektové konstrukce	39
1 Prostředí JShell	40
2 Základní datové typy a jejich literály	58
3 Proměnné	78
4 Základní operátory	91
5 Definice metod	111
6 Ostatní operátory	128
7 Pole	152
8 Rozhodování	166
9 Opakování části kódu	181
Část II: Základní objektové konstrukce	203
10 Základy objektově orientovaného paradigmatu	204
11 Třídy a jejich členy	224
12 Vývojová prostředí a vytvoření aplikace	248
13 Balíčky a knihovny	263
14 Dokumentace API	285
15 Konstrukce interface	298
16 Podrobnosti o konstruktorech	316
17 Úvod do dědění implementace	333
18 Viditelnost členů tříd	351
19 Virtuální metody a jejich přebíjení	369
20 Abstraktní třídy	381

Část III: Pokročilé objektové konstrukce	393
21 Výjimky a aserce	394
22 Generické datové typy a metody	422
23 Typové parametry a argumenty	442
24 Interní datové typy	464
25 Výčtové typy	480
26 Lambda-výrazy	494
27 Anotace	505
28 Vlákna a paralelní procesy	519
28 Moduly.....	526
Literatura.....	551
Rejstřík	552

Podrobný obsah

Stručný obsah	6
Podrobný obsah	8
Seznam výpisů programů	22
Seznam obrázků	27
Seznam odboček – podšeděných bloků	29
Úvod	30
Komu je kniha určena	30
Koncepte výkladu	31
Rozdělení textu	31
Terminologie	32
Použité nástroje	32
Vývojová sada JDK 9	32
Vývojové prostředí JShell	32
Samostatné vývojové prostředí	33
Doprovodné programy	33
Problémy s klávesnicí	33
Historie rozložení České (QWERTY)	34
Syntaktické definice a diagramy	35
Použité typografické konvence	35
Odbočka – podšeděný blok	37
Zpětná vazba	38

Část I: Neobjektové konstrukce 39

1 Prostředí JShell	40
1.1 Charakteristika programu a prostředí JShell	40
1.2 Příprava programu JShell a první spuštění	41
Dávkové soubory pro Windows	41
Po spuštění	42
1.3 Úryvky (snippets)	44
Použití proměnných	45
Identifikace úryvků	45
Středník	45
Více objektů na řádku, zavlečené chyby	45
1.4 Příkazy (commands)	47
Vyloučení úryvku: /drop	47
Přehled aktivních úryvků: /list	47
Přehled aktivních úryvků: /list -all	47
Přehled objektů daného druhu	49
Uložení aktivních úryvků: /save <file>	49

	Uložení všech zadaných úryvků: /save -all <file>.....	50
	Uložení dosavadního průběhu seance: /save -history <file>	50
	Načtení skriptu: /open <file>.....	50
	Ukončení seance: /exit.....	51
	Restart: /reset.....	51
	Znovuzavedení: /reload -restore.....	51
	Natavení startovního skriptu: /set -start <file>.....	51
	Nápověda: /?.....	51
1.5	Základní syntaktická pravidla.....	52
	Bílé znaky	52
	Komentáře	53
1.6	Ovládání	54
	Použití editoru	54
	Nastavení vlastního editoru.....	57
1.7	Záznamy lekcí	57
1.8	Shrnutí.....	57
2	Základní datové typy a jejich literály	58
2.1	Datové typy	58
	Dělení datových typů.....	59
	Primitivní datové typy.....	60
	Objektové datové typy.....	61
	Odkazy na objekty	62
2.2	Literály.....	62
	Literály typu boolean.....	62
	Literály typu int.....	62
	Historická vsuvka – číselné soustavy	63
	Názvy skupin bitů	64
	Literály typu long.....	66
	Literály typu byte a short.....	66
	Literály typu double.....	66
	Celé číslo s příponou	67
	Obyčejné desetinné číslo.....	67
	Číslo v semilogaritmicím tvaru	67
	Literály typu float.....	68
	Literály typu char.....	69
	Prázdný odkaz null.....	72
	Literály typu String.....	73
	Literály typu Class.....	75
2.3	Ještě trocha terminologie.....	75
2.4	Nestandardní hodnoty reálných typů	76
2.5	Shrnutí.....	76
3	Proměnné.....	78
3.1	Pravidla pro tvorbu identifikátorů	78
	Používání znaku \$	79
	Konvence pro velikost písmen.....	79
3.2	Druhy typování	80
	Statické × dynamické typování.....	80
	Definice × odvození datového typu.....	81
	Silné (přísné) × slabé typování	81
	Shrnutí.....	82
3.3	Definice × deklarace.....	82
3.4	Deklarace proměnných	83
3.5	Středníky	84
3.6	Současná deklarace více proměnných.....	85

	Reakce prostředí JShell.....	85
3.7	Redeklarace proměnných v JShell.....	87
3.8	Deklarace s přiřazením počáteční hodnoty.....	87
	Pozor na velikost znaků.....	88
	Zpět k deklaraci s přiřazením počáteční hodnoty.....	89
3.9	Syntaktický diagram.....	90
3.10	Shrnutí.....	90
4	Základní operátory.....	91
4.1	Nejprve trocha teorie.....	91
4.2	Operátor přiřazení =.....	92
	Přiřazení je výraz.....	92
4.3	Unární + a -.....	93
4.4	Aritmetické operátory + - * / %.....	94
	Operátor sčítání.....	94
	Sčítání textových řetězců.....	94
	Operátor odčítání.....	95
	Operátor násobení.....	95
	Operátor dělení.....	95
	Operátor zbytku po dělení.....	96
4.5	Kulaté závorky ().....	96
	Alternativní řešení.....	98
4.6	Operátor přetypování (typ).....	98
	Implicitní přetypování.....	98
	Příklady implicitního přetypování.....	99
	Explicitní přetypování.....	101
	Priorita.....	101
	Kontrola.....	102
	Explicitní přetypování hodnot primitivních typů.....	102
	Příklady.....	103
	Přetypování instancí objektových datových typů.....	105
	Univerzální „přetypování“ na String.....	105
	Textový podpis.....	107
4.7	Specifika číselných typů.....	107
	Malé celočíselné typy.....	107
	Ztráta přesnosti.....	109
	Pořadí vyhodnocování.....	109
	První příklad.....	109
	Druhý příklad.....	110
4.8	Shrnutí.....	110
5	Definice metod.....	111
5.1	Historické ohlédnutí.....	111
5.2	Definice a volání metody.....	112
5.3	Metody s parametry.....	115
	Formální versus skutečné parametry, argumenty.....	115
	Více parametrů.....	116
5.4	Metody vracějící hodnotu.....	117
5.5	Přetěžování metod.....	118
5.6	Lokální proměnné metod.....	119
	Parametry jako lokální proměnné.....	121
	Příklady.....	121
	Jídlna.....	121
	Návratová hodnota.....	121
	Definice metod v editoru.....	122
5.7	Metody s proměnným počtem argumentů.....	123

5.8	Zásobník návratových adres (ZNA)	124
	Parametry × lokální proměnné	125
	Předávání hodnot parametrů	125
	Životnost lokálních proměnných	125
5.9	Přehled definovaných metod	126
5.10	Syntaktický diagram	126
5.11	Shrnutí	127
6	Ostatní operátory	128
6.1	Inkrementační a dekrementační operátory ++ --	128
6.2	Porovnávací operátory < <= == != >= >	130
	Testování shody desetinných čísel	130
	Zvláštnosti porovnávání textových řetězců – stringů	131
	p12 == false	132
	p13 == true	132
	p23 == false	133
	Porovnávání objektů reprezentujících hodnotu	133
6.3	Logické operátory ! & && 	134
6.4	Bitové operátory ~ & ^ << >> >>>	136
6.5	Složené přiřazovací operátory Op=	139
	Příklady využití přetytování	140
6.6	Ternární operátor :? – podmíněný výraz	140
	Ještě jednou porovnávání reálných čísel	142
6.7	Operátor instanceof	144
6.8	Zbývající operátory: new [] ()	146
	Operátor new	146
	Operátor []	147
	Operátor . (tečka)	148
	Operátor volání metody ()	148
6.9	Priorita, asociativita a komutativita operátorů	148
	Priorita	148
	Asociativita	150
	Komutativita	150
6.10	Shrnutí	151
7	Pole	152
7.1	Strukturovaný datový typ – kontejner – pole	152
7.2	Deklarace a inicializace polí	153
	Syntaxe zděděná od jazyků C/C++	154
7.3	Přiřazení hodnoty poli a přetytování polí	155
7.4	Počet prvků pole	157
7.5	Práce s prvky pole	158
7.6	Vícerozměrná pole – pole polí	159
	Obdélníková pole	160
	Zubatá pole	160
	Inicializace dvourozměrného pole	162
	Inicializace vícerozměrného pole	163
7.7	Proměnný počet argumentů metod	163
7.8	Arrays – knihovna metod pro práci s poli	163
7.9	Pole a moderní programování	164
7.10	Shrnutí	165
8	Rozhodování	166
8.1	Jednoduchý podmíněný příkaz	166
8.2	Blok příkazů (složený příkaz)	168

	Vnořování bloků	169
	Proměnné lokální v bloku	169
8.3	Úplný podmíněný příkaz	173
8.4	Složený podmíněný příkaz	174
8.5	Přepínač	176
8.6	Shrnutí	180
9	Opakování části kódu	181
9.1	Obecný cyklus	181
9.2	Cyklus s ukončovací podmínkou – cyklus do-while	182
9.3	Cyklus s počáteční podmínkou – cyklus while	183
9.4	Cyklus s parametrem – cyklus for	185
	Metody s proměnným počtem argumentů	187
9.5	„Dvojtečkový“ cyklus for (cyklus „for each“)	189
9.6	Vnořování cyklů	191
9.7	Cyklus s prázdným tělem	191
9.8	Nekonečný cyklus	193
9.9	Cyklus s podmínkou uprostřed	194
9.10	Příkaz break s návěštím	195
9.11	Příkaz continue	198
9.12	Rekurze	199
	Princip	199
	Přímá a nepřímá rekurze	200
	Přeplnění zásobníku návratových adres	200
9.13	Shrnutí	202

Část II: Základní objektové konstrukce **203**

10	Základy objektově orientovaného paradigmatu	204
10.1	Předmluva	204
10.2	Trocha historie	205
10.3	Motivace OOP	206
10.4	Objekty	206
	Členy objektů	207
10.5	Třídy a jejich instance	207
10.6	Třída jako objekt	208
10.7	Členy třídy a jejich instancí	209
	Přežívající lokální proměnné	210
10.8	Zprávy	210
10.9	Metody	211
10.10	Entity	212
10.11	Polymorfismus, rozhraní, interfejs	212
	Rozhraní × implementace	213
	Atributy × vlastnosti	213
	Vlastnosti v knihovně/platformě/frameworku JavaFX	214
	Signatura × kontrakt	214
	Rozhraní × interface	215
	Interfejs a jeho instance	216
10.12	Objektové datové typy	217
10.13	Dědění	217
	Přirozené (nativní) dědění	218
	Dědění typu (rozhraní)	218
	Dědění implementace	219

	Problémy s děděním – substituční princip Liskové (LSP)	219
10.14	Vlastní instance třídy a mateřská třída objektu	221
10.15	Tři základní principy OOP	221
10.16	Jazyk UML	222
10.17	Shrnutí	223
11	Třídy a jejich členy	224
11.1	Nejjednodušší definice třídy	224
11.2	Konstruktory	225
	Implicitní konstruktor	225
	Vlastní konstruktor a skrytý parametr <code>this</code>	225
	Proč se liší podpisy	227
	Definice tříd jako úryvky	227
11.3	Třída se všemi členy	228
	Statické (třídní) členy	228
	Instanční členy	230
	Konstrukce objektů	230
11.4	Kvalifikace posílaných zpráv	231
	Implicitní kvalifikace	232
11.5	Přetěžování konstruktorů	233
	Kvalifikace klíčovým slovem <code>this</code>	236
11.6	Modifikátory přístupu a skrývání implementace	237
	Veřejné a „neveřejné“ datové typy	238
11.7	Přístupové metody	238
11.8	Modifikátor <code>final</code>	240
	Konstantní atributy	240
	Konstanty vyhodnotitelné v době překladu	240
	Konstantní lokální proměnné	241
	Efektivní konstanty	241
	Zveřejňování konstantních atributů	241
	Modifikátor <code>final</code> v procesu dědění	241
	Neměnnost objektů	242
11.9	Primitivní a obalové datové typy – autoboxing	242
	Převody textových řetězců na hodnoty primitivních typů	243
11.10	Důležité metody klíčových tříd	244
	Třída <code>Object</code>	244
	<code>Object clone()</code>	245
	Mělké a hluboké kopie objektů	245
	<code>boolean equals(Object)</code>	245
	<code>Class<?> getClass()</code>	245
	<code>int hashCode()</code>	246
	<code>String toString()</code>	246
	Třída <code>String</code>	246
	Třída <code>Class</code>	246
	<code>boolean equals(Object)</code>	246
	<code>String getName()</code>	247
	<code>String getSimpleName()</code>	247
	<code>String toString()</code>	247
11.11	Shrnutí	247
12	Vývojová prostředí a vytvoření aplikace	248
12.1	IDE	248
	BlueJ a BlueJ++	249
	Nejpoužívanější IDE	249
12.2	Instalace a spuštění NetBeans	250
12.3	Vytvoření spustitelného projektu v NetBeans	250

	Vytvoření nového projektu	250
	Vytvoření nové třídy	252
	Definice hlavní metody	254
12.4	Překlad a sestavení projektu	255
	Překlad	255
	Sestavení	256
	Spustitelnost JAR-souboru	256
	Spuštění aplikace	257
12.5	Zobrazování varovných hlášení	259
	Zapnutí podrobných hlášení	260
	Doporučení	261
	Vypnutí konkrétního hlášení	261
	Proč vypínat varování	262
12.6	Shrnutí	262
13	Balíčky a knihovny	263
13.1	Velké programy a jejich problémy	263
13.2	Balíčky	264
	Umístění zdrojových souborů	265
	Kořenový (implicitní, defaultní, nepojmenovaný) balíček	265
	Podbalíčky	266
	Konvence pro názvy balíčků	266
	Balíčky doprovodných programů a knihoven	267
13.3	Balíčky a NetBeans	268
13.4	Rozšiřujeme strom balíčků	269
	Názvy tříd	271
13.5	Explicitní ukončení aplikace	271
13.6	Příkaz import	273
	Import zadaného datového typu	273
	Import všech typů ze zadaného balíčku	274
	Podpora zadávání příkazu import ve vývojových prostředích	274
	Výjimčnost balíčku java.lang	275
13.7	Příkaz import static	275
13.8	Syntaktický diagram	276
13.9	Používání knihoven	276
13.10	Typy se stejným názvem v různých balíčcích	279
	Shrnutí	281
13.11	Použití knihovny v JShell	281
	Nastavení proměnné classpath	282
	Nastavení importů	282
	Násilné ukončení aplikace	283
13.12	Shrnutí	284
14	Dokumentace API	285
14.1	Dokumentační komentáře a API	285
14.2	Proč psát srozumitelné a komentované programy	286
	POBLIOCHA	287
14.3	Jak psát dokumentační komentáře	288
14.4	Pomocné značky pro tvorbu dokumentace	289
14.5	Dokumentace balíčku a modulu	290
14.6	Vytvoření a zobrazení dokumentace	292
14.7	Struktura dokumentace API	293
	Práce s panely	294
	Struktura dokumentace datového typu	294
14.8	Zpřehlednění programu	295

14.9	Zakomentování a odkomentování části programu	297
14.10	Shrnutí	297
15	Konstrukce interface	298
15.1	Definice typického interfejsu	298
	Deklarace abstraktních metod	299
	Příklad	299
15.2	Implementace interfejsu třídou	300
15.3	Interfejs se všemi přípustnými typy členů	302
	Motivace k rozšíření – implicitní metody	303
	Statické členy	303
	Instanční členy	305
15.4	Dědění interfejsů	305
15.5	Příklad	305
	Plynulé posuny	306
	Plynulé změny velikosti	307
	Sloučení knihoven	307
15.6	Výhody implicitních metod při návrhu architektury	309
15.7	Řešení kolizí	309
15.8	Specifikace zdroje použité metody	310
	Možné problémy	312
15.9	Speciální interfejsy	313
	Značkovací interfejsy	313
	java.lang.Cloneable	313
	java.io.Serializable	314
	Současné trendy a doporučení	314
	Funkční interfejsy	314
	Interfejs Iterable	314
15.10	Shrnutí	315
16	Podrobnosti o konstruktorech	316
16.1	Opakování: co víme o konstruktorech instancí	316
16.2	Zavádění třídy – java.lang.ClassLoader	317
16.3	Statický konstruktor – konstruktor třídy	318
	Konstruktor interfejsu	318
16.4	Instanční inicializační blok	319
16.5	Dvě části těla konstruktora instancí	319
16.6	Příklad	320
	Konstruktor třídy	326
	3 až 9: Úvodní statický inicializační blok	326
	25: Předčasné použití atributu	327
	8: Nekorektní použití metod	327
	42: Předčasné použití konstanty	327
	62: Nekorektní volání konstrukturu	328
	Inicializační část konstrukturu instancí	328
	12 až 15: Úvodní instanční inicializační blok	328
	149: Deklarace konstanty loaded	329
	153 až 157: Inicializační výpočet	329
	165: Použití this v inicializaci	329
	266 až 269: Závěrečný inicializační blok	329
	Těla konstrukturu instancí	329
	177 až 182: Bezparametrický konstruktor	330
	190 až 196: Jednoperametrický konstruktor	330
	199 až 204: Dvoperametrický konstruktor	330
	213 až 226: Tříperametrický konstruktor	330
16.7	Experimenty	331

16.8	Doporučení	332
	Jediný statický inicializační blok	332
	Bez instančních inicializačních bloků	332
	Inicializovat všechny atributy jednotně	332
16.9	Shrnutí.....	332
17	Úvod do dědění implementace	333
17.1	Úvodní poznámky.....	334
17.2	Definice dceřiné třídy	334
17.3	Rodičovský podobjekt	336
17.4	Konstruktor	337
	Dědění implementace od více rodičů	338
	Konstrukce rodičovského podobjektu.....	338
17.5	Přetížené verze konstruktorů – použití super × this	340
17.6	Konstruktory rodiče a potomka	342
17.7	Demonstrace chování konstruktorů	343
	Definice třídy Graddaughter.....	343
	Provedení akce před příkazem this() nebo super().....	344
	Definice metody constructorReport(Object,Class).....	346
	Spuštění testu	347
	Zavedení třídy.....	347
	Tisk nehotových objektů	347
	Preference vlastních metod.....	348
	Dokončení testu.....	348
	Rodičovský podobjekt je abstrakce	348
17.8	Zákaz vytváření potomků třídy	350
18	Viditelnost členů tříd	351
18.1	Úpravy použitého projektu	351
18.2	Trocha terminologie	352
	Posílání zpráv a volání metod	352
	Přetěžování×přebíjení×zakrývání×přepisování×předefinování metod	352
	Přetěžování metod	352
	Přebíjení metod	353
	Zakrývání metod	353
	Přepsání či předefinování metod.....	353
18.3	Chráněné členy – modifikátor přístupu protected	354
	Shrnutí	356
18.4	Dědění metod	357
	Zděděné, dále neupravované metody	357
	Zděděné metody, pro něž potomek definuje „lepší“ implementaci	358
	Kompatibilita signatur.....	358
18.5	Zakrývání metod předka (method hiding)	359
18.6	Metody, které není možno v potomku zakrýt či přebít – modifikátor final	362
18.7	Zakrývání atributů předka	363
18.8	Metody nově definované v potomku	365
	Staticky × dynamicky typované jazyky	366
	Proč je situace jednoduchá jen zdánlivě	367
	Anotace @Override	367
18.9	Závěr	367
18.10	Shrnutí.....	368
19	Virtuální metody a jejich přebíjení	369
19.1	Princip	369
	Časná a pozdní vazba	370
	Virtuální metody	370

19.2	Které metody jsou v Javě virtuální	371
19.3	Chování virtuálních metod.....	372
19.4	Zdokonalení třídy Square	374
	Přebíání metody copy()	374
	Problémy s nastavováním velikosti	375
	První návrh definice metody setSize(int, int)	376
	Test prvního návrhu	377
	Oprava.....	378
19.5	Co se nám na dědění nelíbí	379
19.6	Shrnutí.....	380
20	Abstraktní třídy	381
20.1	Abstraktní třídy a jejich role v dědické hierarchii	381
	Vytváříme hybrida.....	382
	Abstraktní třída bez abstraktních metod	383
20.2	Konstruktor abstraktní třídy	383
20.3	Deklarace a implementace abstraktních metod.....	384
20.4	Účel abstraktních tříd	386
20.5	Proč společný rodič	386
20.6	Účel abstraktních metod	387
20.7	Návrhový vzor Šablonová metoda (Template method)	388
	Princip.....	388
	Implicitní metody interfejsů	388
	Architektura balíčku eu.pedu.lib17w.geom.....	389
	Metoda toString()	390
20.8	Shrnutí.....	392

Část III: Pokročilé objektové konstrukce 393

21	Výjimky a aserce	394
21.1	Co to jsou výjimky.....	395
21.2	Analyza chybové zprávy	395
	Oznámení o chybě	395
	Jak chyba vznikla – výpis zásobníku návratových adres.....	396
21.3	Nejdůležitější výjimky	397
21.4	Vyhození výjimky	399
	Oddělené vytvoření výjimky	400
21.5	Výjimky a nedosažitelný kód	401
21.6	Co výjimky umí	401
21.7	Hierarchie dědění výjimek	402
21.8	Zachycení vyhozené výjimky	404
	Chování metody exceptionCatching(int)	405
21.9	Syntaktický diagram bloku try ... catch	406
	Několik současně odchyťovaných výjimek	406
	Společná reakce na několik výjimek	407
	Společný úklid – blok finally.....	407
	Příklad	409
21.10	Definice vlastních výjimek	409
21.11	Kontrolované výjimky	411
21.12	Převedení kontrolované výjimky na nekontrolovanou	413
21.13	Informace o skutečném původci výjimky	414
21.14	Ověřování podmínek – příkaz assert	416
	Design by Contract	417