

Stručný obsah

Stručný obsah	6
Podrobný obsah	8
Úvod	23
Část I Neobjektové konstrukce	31
1 Prostředí JShell	32
2 Základní datové typy a jejich literály	51
3 Proměnné	72
4 Základní operátory	85
5 Definice metod	104
6 Ostatní operátory	121
7 Pole	148
8 Rozhodování	164
9 Opakování části kódu	180
Část II Základní objektové konstrukce	201
10 Základy objektově orientovaného paradigmatu	202
11 Třídy a jejich členy	221
12 Vývojová prostředí a vytvoření aplikace	243
13 Balíčky a knihovny	262
14 Dokumentace API	283
15 Konstrukce interface	296
16 Podrobnosti o konstruktorech	314
17 Úvod do dědění implementace	331
18 Viditelnost členů tříd	348
19 Virtuální metody a jejich přebíjení	364
20 Abstraktní třídy	375

Část III Pokročilé objektové konstrukce	387
21 Výjimky a aserce	388
22 Generické datové typy a metody	415
23 Typové parametry a argumenty	435
24 Interní datové typy	456
25 Výčtové typy – třídy typu enum	471
26 Záznamy – třídy typu record.....	485
27 Lambda-výrazy	499
28 Anotace	509
29 Vlákna a paralelní procesy	522
30 Moduly.....	530
Část IV Přílohy	553
A Omezení JVM	554
B Tvorba jednoduchého GUI.....	555
Literatura.....	569
Rejstřík.....	571

Podrobný obsah

Stručný obsah	6
Podrobný obsah	8
Úvod	23
Komu je kniha určena	23
Koncepce výkladu	24
Rozdělení textu	24
Terminologie	25
Použité nástroje	25
Vývojová sada JDK	25
Vývojové prostředí <i>JS</i> hell	25
Samostatné vývojové prostředí	26
Doprovodné programy	26
Předběžné definice nových konstrukcí	26
Předběžná funkce/konstrukce/vlastnost (Preview Feature)	27
Experimentální funkce/konstrukce/vlastnost (Experimental Feature)	27
Inkubační funkce/konstrukce/vlastnost (Incubating Feature, Incubator)	27
Syntaktické definice a diagramy	27
Použité typografické konvence	28
Odbočka – podšeděný blok	30
Zpětná vazba	30

Část I Neobjektové konstrukce 31

1 Prostředí <i>JS</i> hell	32
1.1 Nejprve trochu terminologie	32
Objekt	32
Třída, datový typ	33
Proměnná	33
Atributy × metody	33
1.2 Charakteristika programu a prostředí <i>JS</i> hell	33
1.3 Problémy s klávesnicí	33
1.4 Příprava programu <i>JS</i> hell a první spuštění	34
Dávkové soubory pro <i>Windows</i>	34
Po spuštění	36
1.5 Úryvky (snippets)	37
Použití proměnných	38
Identifikace úryvků	38
Středník	38
Více objektů na řádku, zavlečené chyby	38
1.6 Příkazy (commands)	40
Vyloučení úryvku: /drop	40
Přehled aktivních úryvků: /list	40

Přehled všech úryvků: /list -all	41
Přehled objektů daného druhu	42
Uložení aktivních úryvků: /save <file>	43
Uložení všech zadaných úryvků: /save -all <file>	43
Uložení dosavadního průběhu seance: /save -history <file>	43
Načtení skriptu: /open <file>	43
Ukončení seance: /exit	43
Restart: /reset	44
Znovuzavedení: /reload -restore	44
Nastavení startovního skriptu: /set -start <file>	44
Nápověda: /?	44
1.7 Základní syntaktická pravidla	45
Bílé znaky	45
Komentáře	45
1.8 Ovládání	47
Použití editoru	47
Nastavení vlastního editoru	49
1.9 Doprovodné programy	50
1.10 Shrnutí	50
2 Základní datové typy a jejich literály	51
2.1 Datové typy	51
Dělení datových typů	52
Primitivní datové typy	53
Objektové datové typy	54
Odkazy na objekty	54
2.2 Literály	55
Literály typu boolean	55
Literály typu int	55
Historická vsuvka – číselné soustavy	56
Názvy skupin bitů	57
Literály typu long	57
Literály typu byte a short	58
Literály typu double	58
Celé číslo s příponou	59
Obyčejné desetinné číslo	59
Číslo v exponentovém tvaru	59
Literály typu float	61
Literály typu char	61
Prázdný odkaz null	65
Literály typu String	65
Textové bloky	67
Literály typu Class	69
2.3 Ještě trocha terminologie	71
2.4 Nestandardní hodnoty reálných typů	71
2.5 Shrnutí	71
3 Proměnné	72
3.1 Pravidla pro tvorbu identifikátorů	72
Používání znaku \$	73
Konvence pro velikost písmen	73
3.2 Druhy typování	74
Statické × dynamické typování	74
Definice × odvození datového typu	75
Silné (přísné) × slabé typování	75
Shrnutí	76

3.3	Definice × deklarace.....	76
3.4	Deklarace proměnných.....	76
3.5	Středníky.....	79
3.6	Současná deklarace více proměnných.....	79
	Reakce prostředí <i>JShell</i>	79
3.7	Redeklarace proměnných v <i>JShell</i>	81
3.8	Deklarace s přiřazením počáteční hodnoty.....	81
	Pozor na velikost znaků.....	82
	Zpět k deklaraci s přiřazením počáteční hodnoty.....	83
3.9	Syntaktický diagram.....	83
3.10	Shrnutí.....	84
4	Základní operátory.....	85
4.0	Inicializace prostředí <i>JShell</i>	85
4.1	Nejprve trochu teorie.....	86
4.2	Operátor přiřazení =.....	87
	Přiřazení je výraz.....	87
4.3	Unární + a -.....	88
4.4	Aritmetické operátory + - * / %.....	88
	Operátor sčítání.....	88
	Sčítání stringů.....	88
	Operátor odčítání.....	89
	Operátor násobení.....	90
	Operátor dělení.....	90
	Operátor zbytku po dělení.....	90
4.5	Kulaté závorky ().....	91
	Alternativní řešení.....	92
4.6	Operátor přetypování (<i>typ</i>).....	92
	Implicitní přetypování.....	92
	Příklady implicitního přetypování.....	93
	Explicitní přetypování.....	94
	Priorita.....	95
	Kontrola.....	95
	Explicitní přetypování hodnot primitivních typů.....	96
	Příklady.....	96
	Přetypování instancí objektových datových typů.....	98
	Univerzální „přetypování“ na <i>String</i>	99
	Textový podpis.....	100
4.7	Specifika číselných typů.....	100
	Malé celočíselné typy.....	100
	Ztráta přesnosti.....	102
	Pořadí vyhodnocování.....	102
	První příklad.....	102
	Druhý příklad.....	103
4.8	Shrnutí.....	103
5	Definice metod.....	104
5.1	Historické ohlédnutí.....	104
5.2	Definice a volání metody.....	105
5.3	Volání metody.....	107
5.4	Metody s parametry.....	108
	Parametry versus argumenty.....	109
	Více parametrů.....	109
	Předávání hodnot parametrů.....	110
5.5	Metody vracějící hodnotu.....	111
5.6	Přetěžování metod.....	112

5.7 Lokální proměnné metod.....	113
Postup volání metody	114
Zásobník návratových adres – ZNA.....	115
Parametry × lokální proměnné	115
Životnost lokálních proměnných	115
5.8 Příklady	116
Jídlna.....	116
Návratová hodnota	116
Definice metod v editoru.....	117
5.9 Metody s proměnným počtem argumentů.....	118
5.10 Přehled definovaných metod	119
5.11 Syntaktický diagram	119
5.12 Shrnutí.....	120
6 Ostatní operátory	121
6.1 Inkrementační a dekrementační operátory ++ --	121
6.2 Porovnávací operátory < <= == != >= >	123
Testování shody desetinných čísel	124
Zvláštnosti porovnávání stringů	125
p12 == false.....	126
p13 == true.....	126
p23 == false.....	127
Porovnávání objektů reprezentujících hodnotu	127
6.3 Logické operátory ! & && 	127
6.4 Bitové operátory ~ & ^ << >> >>>	130
6.5 Složené přiřazovací operátory Op=.....	133
Příklady využití přetypování	133
6.6 Ternární operátor :? – podmíněný výraz.....	134
Ještě jednou porovnávání reálných čísel	136
6.7 Přepínač – výraz switch	137
Pravidla.....	138
Příklad.....	138
6.8 Operátor instanceof	139
Shoda vzorů pro operátor instanceof.....	141
6.9 Zbylé operátory: new [] ()	142
Operátor new	142
Operátor []	144
Operátor . (tečka).....	144
Operátor volání metody ()	144
6.10 Priorita, asociativita a komutativita operátorů	145
Priorita	145
Asociativita.....	146
Komutativita	147
6.11 Shrnutí.....	147
7 Pole.....	148
7.0 Představení kapitoly	148
7.1 Strukturovaný datový typ – kontejner – pole	149
7.2 Deklarace a inicializace polí	150
Syntaxe zděděná od jazyků C/C++.....	151
7.3 Přiřazení hodnoty poli a přetypování polí.....	151
7.4 Počet prvků pole	153
7.5 Práce s prvky pole.....	154
7.6 Vícerozměrná pole – pole polí.....	155
Obdélníková pole.....	156

Zubatá pole.....	157
Inicializace dvourozměrného pole.....	159
Inicializace vícerozměrného pole.....	159
7.7 Proměnný počet argumentů metod.....	160
7.8 Arrays – knihovna metod pro práci s poli.....	160
7.9 Emulace předání argumentu odkazem.....	161
7.10 Pole a moderní programování.....	162
7.11 Shrnutí.....	163
8 Rozhodování.....	164
8.1 Jednoduchý podmíněný příkaz.....	164
8.2 Blok příkazů (složený příkaz).....	166
Vnořování bloků.....	167
Proměnné lokální v bloku.....	167
8.3 Úplný podmíněný příkaz.....	170
8.4 Složený podmíněný příkaz.....	171
8.5 Přepínač – příkaz switch.....	174
Pravidla.....	174
Příklad.....	176
8.6 Shrnutí.....	179
9 Opakování části kódu.....	180
9.1 Obecný cyklus.....	180
9.2 Cyklus s ukončovací podmínkou – cyklus do...while.....	181
9.3 Cyklus s počáteční podmínkou – cyklus while.....	182
9.4 Cyklus s parametrem – cyklus for.....	184
Metody s proměnným počtem argumentů.....	186
9.5 „Dvojtečkový“ cyklus for (cyklus „for each“).....	188
9.6 Vnořování cyklů.....	189
9.7 Cyklus s prázdným tělem.....	191
9.8 Nekonečný cyklus.....	192
9.9 Cyklus s podmínkou uprostřed.....	193
9.10 Příkaz break s návěstím.....	194
9.11 Příkaz continue.....	197
9.12 Rekurze.....	198
Princip.....	198
Přímá a nepřímá rekurze.....	198
Přeplnění zásobníku návratových adres.....	199
9.13 Shrnutí.....	200

Část II Základní objektové konstrukce 201

10 Základy objektově orientovaného paradigmatu.....	202
10.1 Předmluva.....	202
10.2 Trocha historie.....	203
10.3 Motivace OOP.....	203
10.4 Objekty.....	204
Členy objektů.....	204
10.5 Třídy a jejich instance.....	205
10.6 Třída jako objekt.....	206
10.7 Členy třídy a jejich instancí.....	207
Přežívající lokální proměnné.....	207
10.8 Zprávy.....	208
10.9 Metody.....	208

10.10	Entity	209
10.11	Polymorfismus, rozhraní, interfejs	209
	Rozhraní x implementace	210
	Atributy x vlastnosti	211
	Vlastnosti v knihovně/platformě/frameworku <i>JavaFX</i>	211
	Signatura x kontrakt	212
	Rozhraní x interface	212
	Interfejs a jeho instance	213
10.12	Objektové datové typy	213
10.13	Dědění	214
	Přirozené (nativní) dědění	215
	Dědění typu (rozhraní)	215
	Dědění implementace	216
	Problémy s děděním – substituční princip Liskové (LSP)	216
10.14	Vlastní instance třídy a mateřská třída objektu	217
10.15	Tři základní principy OOP	217
10.16	Jazyk UML	218
10.17	Správa paměti	219
10.18	Shrnutí	220
11	Třídy a jejich členy	221
11.1	Nejjednodušší definice třídy	221
11.2	Konstruktory	222
	Implicitní konstruktor	222
	Vlastní konstruktor a skrytý parametr <i>this</i>	222
	Proč se liší podpisy	223
	Definice tříd jako úryvky	224
11.3	Třída se všemi členy	224
	Statické (třídní) členy	225
	Instanční členy	226
	Konstrukce objektů	227
11.4	Kvalifikace posílaných zpráv	227
	Implicitní kvalifikace	228
11.5	Přetěžování konstruktorů	229
	Kvalifikace klíčovým slovem <i>this</i>	232
11.6	Modifikátory přístupu a skrývání implementace	233
	Veřejné a „neveřejné“ datové typy	234
11.7	Přístupové metody	234
11.8	Modifikátor <i>final</i>	235
	Konstantní atributy	236
	Konstanty vyhodnotitelné v době překladu	236
	Konstantní lokální proměnné	236
	Efektivní konstanty	237
	Zveřejňování konstantních atributů	237
	Modifikátor <i>final</i> v procesu dědění	237
	Neměnnost objektů	237
11.9	Primitivní a obalové datové typy – autoboxing	238
	Převody stringů na hodnoty primitivních typů	239
11.10	Důležité metody klíčových tříd	240
	Třída <i>Object</i>	240
	<i>Object clone()</i>	240
	Mělké a hluboké kopie objektů	240
	<i>boolean equals(Object)</i>	241
	<i>Class<?> getClass()</i>	241
	<i>int hashCode()</i>	241
	<i>String toString()</i>	241

void finalize()	241
Třída String	242
boolean equals(Object)	242
Třída Class	242
boolean equals(Object)	242
String getName()	242
String getSimpleName()	242
String toString()	242
11.11 Shrnutí	242
12 Vývojová prostředí a vytvoření aplikace	243
12.1 IDE	243
BlueJ a BlueJ++	244
Nejpoužívanější IDE	244
12.2 Instalace a spuštění <i>NetBeans</i>	245
12.3 Vytvoření spustitelného projektu v <i>NetBeans</i>	245
Vytvoření nového projektu	245
Vytvoření nové třídy	247
Definice hlavní metody	249
12.4 Překlad a sestavení projektu	250
Nastavování parametrů překladu	251
Nový překlad a sestavení	252
Sestavení	253
12.5 Spuštění aplikace	253
Spustitelnost JAR-souboru – hlavní třída aplikace	253
Nastavení parametrů virtuálního stroje	255
Soubor MANIFEST.MF	255
Spuštění aplikace z příkazového řádku	256
Syntaktický diagram spouštění aplikace	257
Java	257
ArgumentVM	257
Spouštěná třída	257
ArgumentProgramu	258
Příklady	258
12.6 Zobrazování varovných hlášení	259
Doporučení	259
Vypnutí konkrétního hlášení	259
Proč vypínat varování	260
12.7 Shrnutí	261
13 Balíčky a knihovny	262
13.1 Velké programy a jejich problémy	262
13.2 Balíčky	263
Umístění zdrojových souborů	264
Kořenový (implicitní, defaultní, nepojmenovaný) balíček	264
Podbalíčky	265
Konvence pro názvy balíčků	265
Balíčky doprovodných programů a knihoven	265
13.3 Balíčky a <i>NetBeans</i>	266
13.4 Rozšiřujeme strom balíčků	267
Názvy tříd	269
13.5 Explicitní ukončení aplikace	270
13.6 Příkaz import	271
Import zadaného datového typu	271
Import všech typů ze zadaného balíčku	272
Podpora zadávání příkazu import ve vývojových prostředích	272
Výjimečnost balíčku java.lang	273

13.7	Příkaz <code>import static</code>	273
13.8	Syntaktický diagram	274
13.9	Používání knihoven.....	274
13.10	Typy se stejným názvem v různých balíčcích	277
	Shrnutí	279
13.11	Použití knihovny v JShell	279
	Nastavení proměnné <code>classpath</code>	280
	Nastavení importů	280
	Násilné ukončení aplikace	281
13.12	Shrnutí.....	282
14	Dokumentace API.....	283
14.1	Dokumentační komentáře a API	283
14.2	Proč psát srozumitelné a komentované programy	284
	POBLOCHA	285
	Jak dokumentační komentáře zobrazovat	286
14.3	Jak psát dokumentační komentáře	286
14.4	Pomocné značky pro tvorbu dokumentace	287
14.5	Dokumentace balíčku a modulu	288
14.6	Vytvoření a zobrazení dokumentace	290
14.7	Struktura dokumentace API	292
	Struktura dokumentace datového typu.....	292
	Rychlé vyhledání	293
14.8	Zpřehlednění programu	293
14.9	Zakomentování a odkomentování části programu	295
14.10	Shrnutí.....	295
15	Konstrukce interface.....	296
15.1	Definice typického interfejsu	296
	Deklarace abstraktních metod.....	297
	Příklad	297
15.2	Implementace interfejsu třídou	298
15.3	Interfejs se všemi přípustnými typy členů.....	300
	Motivace k rozšíření – implicitní metody.....	300
	Statické členy	302
	Instanční členy	302
15.4	Dědění interfejsů	303
15.5	Příklad	303
	Plynulé posuny	303
	Plynulé změny velikosti	304
	Sloučení knihoven	306
15.6	Výhody implicitních metod při návrhu architektury	306
15.7	Řešení kolizí.....	307
15.8	Specifikace zdroje použité metody.....	309
	Možné problémy	309
15.9	Speciální interfejsy.....	311
	Značkovací interfejsy	311
	<code>java.lang.Cloneable</code>	311
	<code>java.io.Serializable</code>	311
	Současné trendy a doporučení.....	311
	Funkční interfejsy	311
	Interfejs <code>Iterable</code>	312
15.10	Shrnutí.....	313

16	Podrobnosti o konstruktorech	314
16.1	Opakování: co víme o konstruktorech instancí	314
16.2	Zavádění třídy – <code>java.lang.ClassLoader</code>	315
16.3	Statický konstruktor – konstruktor třídy	316
	Konstruktor interfejsu	316
16.4	Instanční inicializační blok	317
16.5	Dvě části těla konstrukturu instancí	317
16.6	Příklad	318
	Konstruktor třídy	324
	3–9: Úvodní statický inicializační blok	324
	25: Předčasné použití atributu	324
	8: Nekorektní použití metod	324
	42: Předčasné použití konstanty	324
	62: Nekorektní volání konstrukturu	325
	Inicializační část konstrukturu instancí	325
	12–15: Úvodní instanční inicializační blok	325
	149: Deklarace konstanty <code>loaded</code>	326
	153–157: Inicializační výpočet	326
	165: Použití <code>this</code> v inicializaci	326
	266–269: Závěrečný inicializační blok	326
	Tělo konstruktorů instancí	326
	177–182: Bezparametrický konstruktor	326
	190–196: Jednparametrický konstruktor	327
	205–210: Dvoupametrický konstruktor	327
	220–233: Tříparametrický konstruktor	327
16.7	Experimenty	327
16.8	Doporučení	328
	Jediný statický inicializační blok	328
	Bez instančních inicializačních bloků	328
	Inicializovat všechny atributy jednotně	329
16.9	Skutečný název metody konstrukturu	329
16.10	Shrnutí	330
17	Úvod do dědění implementace	331
17.0	Představení kapitoly	331
17.1	Úvodní poznámky	332
17.2	Definice dceřiné třídy	332
17.3	Rodičovský podobjekt	334
	Dědění implementace od více rodičů	335
17.4	Konstruktor	336
	Konstrukce rodičovského podobjektu	336
17.5	Přetížené verze konstruktorů – použití <code>super</code> × <code>this</code>	338
17.6	Konstruktory rodiče a potomka	340
17.7	Demonstrace chování konstruktorů	340
	Definice třídy <code>Graddaughter</code>	340
	Provedení akce před příkazem <code>this()</code> nebo <code>super()</code>	341
	Definice metody <code>constructorReport(Object, Class)</code>	343
	Spuštění testu	344
	Zavedení třídy	344
	Tisk nehotových objektů	344
	Preference vlastních metod	345
	Dokončení testu	345
	Rodičovský podobjekt je abstrakce	345
17.8	Zákaz vytváření potomků třídy	347
17.9	Shrnutí	347

18 Viditelnost členů tříd	348
18.1 Úpravy použitého projektu	348
18.2 Trocha terminologie	349
Posílání zpráv a volání metod	349
Přetěžování×přebíjení×zakrývání×přepisování×předefinování metod	349
Přetěžování metod	349
Přebíjení metod	349
Zakrývání metod	350
Přepsání či předefinování metod	350
18.3 Chráněné členy – modifikátor přístupu <code>protected</code>	350
Shrnutí	353
18.4 Dědění metod	353
Zděděné, dále neupravované metody	353
Zděděné metody, pro něž potomek definuje „lepší“ implementaci	354
Kompatibilita signatur	354
18.5 Zakrývání metod předka (<code>method hiding</code>)	355
18.6 Metody, které není možno v potomku zakrýt či přebít – modifikátor <code>final</code>	357
18.7 Zakrývání atributů předka	359
18.8 Metody nově definované v potomku	360
Proč je situace jednoduchá jen zdánlivě	361
Anotace <code>@Override</code>	361
Stacky × dynamicky typované jazyky	362
18.9 Závěr	363
18.10 Shrnutí	363
19 Virtuální metody a jejich přebíjení	364
19.1 Princip	364
Časná a pozdní vazba	365
Virtuální metody	365
19.2 Které metody jsou v Javě virtuální	366
19.3 Chování virtuálních metod	366
19.4 Zdokonalení třídy <code>Square</code>	368
Přebíjení metody <code>copy()</code>	369
Problémy s nastavováním velikosti	369
První návrh definice metody <code>setSize(int, int)</code>	370
Test prvního návrhu	371
Oprava	372
19.5 Co se nám na dědění nelíbí	374
19.6 Shrnutí	374
20 Abstraktní třídy	375
20.1 Abstraktní třídy a jejich role v dědické hierarchii	375
Vytváříme hybrida	376
Abstraktní třída bez abstraktních metod	377
20.2 Konstruktor abstraktní třídy	377
20.3 Deklarace a implementace abstraktních metod	378
20.4 Účel abstraktních tříd	380
20.5 Proč společný rodič	380
20.6 Účel abstraktních metod	381
20.7 Návrhový vzor Šablonová metoda (<code>Template method</code>)	381
Princip	381
Implicitní metody interfejsů	382
Architektura balíčku <code>eu.pedu.lib20s.geom</code>	382
Metoda <code>toString()</code>	384
20.8 Shrnutí	386

Část III Pokročilé objektové konstrukce	387
21 Výjimky a aserce	388
21.0 Představení kapitoly	388
21.1 Co to jsou výjimky	389
21.2 Analýza chybové zprávy	389
Oznámení o chybě	389
Jak chyba vznikla – výpis zásobníku návratových adres	390
21.3 Nejdůležitější výjimky	391
21.4 Vyhození výjimky	393
Oddělené vytvoření výjimky	394
21.5 Výjimky a nedosažitelný kód	395
21.6 Co výjimky umí	395
21.7 Hierarchie dědění výjimek	396
21.8 Zachycení vyhozené výjimky	397
Chování metody <code>exceptionCatching(int)</code>	399
21.9 Syntaktický diagram bloku <code>try ... catch</code>	399
Několik současně odchytávaných výjimek	399
Společná reakce na několik výjimek	400
Společný úklid – blok <code>finally</code>	401
Příklad	401
21.10 Definice vlastních výjimek	403
21.11 Kontrolované výjimky	404
21.12 Převedení kontrolované výjimky na nekontrolovanou	406
21.13 Informace o skutečném původci výjimky	407
21.14 Ověřování podmínek – příkaz <code>assert</code>	409
Design by Contract	410
21.15 Kdopak mne to volal	413
21.16 Shrnutí	414
22 Generické datové typy a metody	415
22.1 Motivace	415
22.2 Generické a parametrizované datové typy	418
22.3 Definice generických typů	421
22.4 Použití generických typů	422
22.5 Příklad generických datových typů a očišťování	424
22.6 Rizika nepoužití typových argumentů	424
22.7 Varování překladače a jejich potlačení	427
Varování o použití předběžných funkcí	428
Další varování	429
22.8 Generické metody	429
22.9 Shrnutí	434
23 Typové parametry a argumenty	435
23.1 Omezení typových argumentů	435
Typové argumenty s více předky	436
Vzájemné závislosti typových parametrů	436
23.2 Příklad a očišťování podrobněji	437
Doporučené pořadí omezujících interfejsů	437
Ztráta informace při běhu	439
Přemostovací metody	439
23.3 Zakázané operace	441
Za typové parametry nelze dosazovat primitivní typy	441
Typové parametry třídy není možno použít u statických členů	441
Nelze vytvořit instanci typového parametru	441

Reflexe	442
Nelze vytvořit pole instancí typového parametru ani parametrizovaného typu.....	443
Výjimky	444
23.4 Proměnný počet argumentů – @SafeVarargs	444
Omezení.....	445
Vytvoření pole hodnot.....	446
23.5 Nejednoznačnosti a kolize.....	446
Falešně přetížená metoda.....	446
Nová metoda koliduje se zděděnou.....	447
Kolize požadovaných interfejsů.....	448
Kolize implementovaných interfejsů	449
Potomci a předci generických typů – špatné pochopení dědičnosti.....	450
23.6 Žolíky	451
23.7 Příklad: datový typ <code>Interval<T extends Comparable<? super T>></code>	452
23.8 Shrnutí.....	455
24 Interní datové typy	456
24.1 Motivace	456
Pomocný soukromý typ	457
Objekt znající útroby a implementující veřejné rozhraní	457
Sdružení souvisejících typů	457
24.2 Terminologie.....	458
24.3 Společné charakteristiky interních typů	459
24.4 Globální interní (členské) datové typy	460
24.5 Vnořené datové typy	460
24.6 Vnitřní třídy	461
Interní interfejsy a výčtové typy bez modifikátoru <code>static</code>	461
24.7 Příklad na vnořené a vnitřní třídy	462
Vnořená <code>Elements</code> × vnitřní <code>SAIterator</code>	463
Veřejná <code>Elements</code> × soukromá <code>SAIterator</code>	464
Definice třídy <code>SparseArray.Element</code>	464
Definice třídy <code>SparseArray.SAIterator</code>	465
Definice třídy <code>SparseArrayTest</code>	465
24.8 Lokální třídy	468
Pojmenované lokální třídy	469
Anonymní třídy	469
Použití anonymních tříd.....	470
24.9 Shrnutí.....	470
25 Výčtové typy – třídy typu <code>enum</code>	471
25.1 Nejjednodušší definice	471
Překladačem přidané atributy a metody	473
Atribut <code>\$VALUES</code>	473
<code>public static final NázevTypu[] values()</code>	473
<code>public static NázevTypu valueOf(String name)</code>	473
25.2 Třída <code>Enum</code>	473
Zděděné metody	474
<code>public final String name()</code>	474
<code>public final int ordinal()</code>	474
<code>public final int compareTo(E o)</code>	474
<code>public final Class<E> getDeclaringClass()</code>	475
<code>public static <T extends Enum<T>> T valueOf(Class<T> enumType, String name)</code>	475
Přebité verze metod zděděných od třídy <code>Object</code>	475
<code>protected final Object clone() throws CloneNotSupportedException</code>	475
<code>equals(Object)</code>	475

hashCode()	475
public String toString()	475
Serializace	475
25.3 Použití výčtových typů v programu	476
Přepínač	476
Přidaná anonymní třída	477
Cyklus	478
25.4 Složitější definice výčtových typů	478
25.5 Akční výčtové typy	482
Class-objekty instancí funkčních výčtových typů	483
25.6 Shrnutí	484
26 Záznamy – třídy typu record	485
26.1 Teoretická předehra	485
Odkazové a hodnotové datové typy	486
Proměnné a neměnné hodnotové typy	486
Přepřavky	487
Problémy a motivace	487
Koncepční vlastnosti	488
26.2 Základní syntaxe	488
26.3 Automaticky definované členy	489
26.4 Některé možnosti	489
26.5 Kanonický konstruktor	490
26.6 Možnosti a omezení	492
26.7 Komplexnější definice	493
Prověrka konstruktorů	495
Definované metody	496
Řešení vzniklých problémů	498
26.8 Shrnutí	498
27 Lambda-výrazy	499
27.1 Motivace	499
27.2 Koncepce lambda-výrazů	500
27.3 Funkční interfejsy	500
27.4 Syntaxe lambda-výrazů	503
Jednoduchý příklad	504
Lambda-výrazy zastupující metody	504
Lambda-výraz zastupující konstruktor	505
27.5 Použití lokálních proměnných z okolního bloku	507
27.6 Shrnutí	508
28 Anotace	509
28.1 Co jsou anotace	509
28.2 Označování deklarací anotacemi	510
28.3 Kde všude můžeme anotace použít	512
Anotování balíčků	513
Anotování parametru this	514
28.4 Anotace ve standardní knihovně	514
Standardní anotace v balíčku java.lang	514
@Deprecated	514
@Override	515
@SuppressWarnings	515
@SafeVarargs	515
@FunctionalInterface	516
Standardní anotace v balíčku javax.annotation	516
Metaanotace	516

@Documented.....	516
@Inherited.....	516
@Repeatable.....	517
@Retention.....	517
@Target.....	517
28.5 Syntaxe definice anotací.....	518
Jednoduchá značkovací anotace.....	520
28.6 Získávání informací o anotacích za běhu programu.....	521
28.7 Shrnutí.....	521
29 Vlákna a paralelní procesy.....	522
29.1 Paralelní provádění více činností.....	522
Kooperativní plánování.....	523
Preemptivní plánování.....	523
Použité plánování.....	523
29.2 Vlákna a jejich stavy.....	523
29.3 Sdílení zdrojů.....	525
29.4 Kritické sekce a monitory.....	526
29.5 Synchronizace.....	526
29.6 Uvolnění kritické sekce.....	527
29.7 Jemnější způsoby synchronizace.....	528
Modifikátor <code>volatile</code>	528
Atomické objekty.....	528
29.8 Novější metody práce s vlákny.....	528
29.9 Shrnutí.....	529
30 Moduly.....	530
30.1 Motivace.....	530
Problémy předchozích verzí Javy.....	531
Cíle projektu <i>Jigsaw</i>	532
Dosažené výhody.....	532
30.2 Srovnání instalace Javy 8 a Javy 14.....	533
30.3 Modul × Balíček.....	534
30.4 Soubor <code>module-info.java</code>	535
Syntaktický diagram deklarace modulu.....	535
Název modulu.....	537
Direktiva <code>requires</code>	537
Direktiva <code>exports</code>	538
Direktiva <code>opens</code> a modifikátor <code>open</code>	538
Direktiva <code>uses</code>	538
Direktiva <code>provides</code>	538
30.5 Modulární JAR-soubor.....	539
30.6 Proměnná <code>modulepath</code>	539
30.7 Vytvoření modulární aplikace.....	539
Vytvoření projektu.....	540
Definice modulu.....	541
Přidání zdrojových souborů.....	542
Odstraňování chyb z nepokrytých závislostí.....	543
Přidání modulu <code>eu.pedu.lib20s.geom</code>	545
Přidání modulu <code>eu.pedu.lib20s.canvas</code>	546
Přidání modulu <code>eu.pedu.lib20s.canvasmanager</code>	547
Závěrečná podoba deklarací modulů.....	548
JAR-soubor s více moduly.....	548
30.8 Klasifikace modulů.....	548
Běžný modul (<code>normal module</code>).....	549
Otevřený modul (<code>open module</code>).....	549

Automatický modul (automatic module).....	549
Vlastnosti automatických modulů	551
Nepojmenované moduly (unnamed modules).....	551
30.9 Moduly a platforma <i>JShell</i>	551
30.10 Shrnutí.....	552

Část IV Přílohy 553

A Omezení JVM	554
B Tvorba jednoduchého GUI.....	555
B.1 Trocha historie – přehled rozšířených platforem	555
Platforma AWT	555
Platforma Swing.....	556
Platforma SWT	556
Platforma JavaFX.....	556
B.2 Základní koncepce platforem pro tvorbu GUI.....	557
GUI a využívání vláken	557
Modalita dialogových oken.....	558
B.3 Prostředky pro triviální okenní vstup a výstup.....	558
Třída <code>eu.pedu.lib20s.util.IO</code>	558
<code>void setDialogsPosition(int x, int y)</code>	559
<code>boolean confirm(Object question)</code>	559
<code>int choose(Object question, String... buttons)</code>	559
<code>double enter(Object prompt, double defaultDouble) int</code> <code>enter(Object prompt, int defaultInt) String enter(Object prompt,</code> <code>String defaultString)</code>	559
<code>String select(Object prompt, String... options)</code>	559
<code>void inform(Object text)</code>	559
Třída <code>javax.swing.JOptionPane</code>	559
<code>showMessageDialog</code>	559
<code>showConfirmDialog</code>	559
<code>showOptionDialog</code>	560
<code>showInputDialog</code>	560
B.4 Základy koncepce platforem AWT a Swing.....	560
Komponenty a kontejnery.....	560
Správce rozvržení – <code>LayoutManager</code>	560
<code>BorderLayout</code>	561
<code>BoxLayout</code>	561
<code>FlowLayout</code>	561
Kontejnery – okna a panely	561
<code>JFrame</code>	561
<code>Box</code>	561
<code>JPanel</code>	561
Události a jejich posluchači.....	562
B.5 Příklad.....	562
Literatura.....	569
Rejstřík.....	571