

# knihovna programátora

- Podrobný výklad vlastností jazyka od naprostých základů až po pokročilé, běžně neprobírané konstrukce
- **Vedle aktuálních konstrukcí vysvětluje i ty, které jsou zabudovány jen předběžně a stanou se řádnými až v některé z příštích verzí**
- Pro demonstraci vykládaných konstrukcí bez zbytečného pomocného kódu využívá zabudované REPL prostředí JShell
- Ukazuje, jak efektivně experimentovat a využitím prostředí JShell získat okamžité odpovědi
- Ideální jako učebnice i referenční příručka

RUDOLF  
PECINOVSKÝ

# Java 21

Kompletní příručka jazyka



knihovna programátora

---

RUDOLF  
PECINOVSKÝ

# Java 21

**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 21

## Kompletní příručka jazyka

Vydala Grada Publishing, a.s.

U Průhonu 22, Praha 7

obchod@grada.cz, www.grada.cz

tel.: +420 234 264 401

jako svou 8677. publikaci

Odpovědný redaktor: Petr Somogyi

Grafická úprava a sazba Rudolf Pecinovský

Počet stran 640

První vydání, Praha 2023

Vytiskly Tiskárny Havlíčkův Brod, a.s.

© Grada Publishing, a.s., 2023

Cover Design © Grada Publishing, a. s., 2023

Cover Photo © Depositphotos

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

ISBN 978-80-271-7041-8 (pdf)

ISBN 978-80-247-0599-6 (print)

---

*Všem, kteří se chtějí něco naučit*

# Stručný obsah

Stručný obsah .....	6
Podrobný obsah .....	8
Úvod .....	25
<b>Část I Neobjektové konstrukce</b>	<b>35</b>
<hr/>	
1 Prostředí JShell .....	36
2 Základní datové typy a jejich literály .....	58
3 Proměnné .....	82
4 Základní operátory .....	96
5 Definice metod .....	115
6 Ostatní operátory .....	135
7 Pole .....	165
8 Rozhodování .....	181
9 Opakování části kódu .....	200
<b>Část II Základní objektové konstrukce</b>	<b>221</b>
<hr/>	
10 Základy objektově orientovaného paradigmatu .....	222
11 Třídy a jejich členy .....	246
12 Vytvoření aplikace a vývojová prostředí .....	270
13 Balíčky a knihovny .....	291
14 Dokumentace API .....	310
15 Konstrukce interface .....	322
16 Podrobnosti o konstruktorech .....	340
17 Úvod do dědění implementace .....	357
18 Viditelnost členů tříd .....	380
19 Virtuální metody a jejich přebíjení .....	396
20 Abstraktní třídy .....	407

<b>Část III Pokročilejší objektové konstrukce</b>	<b>419</b>
21 Výjimky a aserce .....	420
22 Generické datové typy a metody .....	447
23 Typové parametry a argumenty .....	466
24 Interní datové typy .....	487
25 Výčtové typy – třídy typu enum .....	503
26 Záznamové třídy – třídy typu record.....	521
27 Další použití rozpoznávání vzorů .....	537
28 Lambda-výrazy .....	552
29 Anotace a šablonové procesory .....	569
30 Vlákna a paralelní procesy .....	584
31 Moduly.....	593
32 Kategorie datových typů, hodnotové typy .....	607
<b>Část IV Přílohy</b>	<b>615</b>
A Tvorba jednoduchého GUI.....	616
Literatura.....	630
Rejstřík.....	632

# Podrobný obsah

Stručný obsah .....	6
Podrobný obsah .....	8
Úvod .....	25
Komu je kniha určena .....	25
Koncepce výkladu .....	25
Rozdělení textu .....	26
Terminologie .....	26
Použité nástroje .....	27
Vývojová sada JDK .....	27
Vývojové prostředí <i>JShell</i> .....	27
Samostatné vývojové prostředí .....	27
Doprovodné programy .....	28
Zlom .....	28
Předběžné definice nových konstrukcí .....	29
Předběžná funkce/konstrukce/vlastnost (Preview Feature) .....	29
Globální nastavení podpory předběžných konstrukcí .....	29
Experimentální funkce/konstrukce/vlastnost (Experimental Feature) .....	30
Inkubační funkce/konstrukce/vlastnost (Incubating Feature, Incubator) .....	30
Syntaktické definice a diagramy .....	30
Použité typografické konvence .....	30
Odbočka – podšeděný blok .....	32
Zpětná vazba .....	32

## Část I Neobjektové konstrukce 35

1 Prostředí <i>JShell</i> .....	36
1.1 Nejprve trocha terminologie .....	36
Objektově orientované paradigma – OO paradigma, OOP .....	37
Objekt v programu .....	37
Třída, datový typ .....	37
Proměnná .....	38
Atributy .....	38
Metody .....	38
Vlastnosti .....	38
Interní typy .....	38
Členy .....	38
1.2 Charakteristika programu a prostředí <i>JShell</i> .....	38
1.3 Problémy s klávesnicí .....	39
1.4 Příprava programu <i>JShell</i> a první spuštění .....	39
Dávkové soubory pro <i>Windows</i> .....	40
Po spuštění .....	41
1.5 Úryvky (snippets) .....	42
Použití proměnných .....	43



Identifikace úryvků .....	44
Středník .....	44
Více objektů na řádku, zavlečené chyby .....	44
<b>1.6 Příkazy (commands) .....</b>	<b>45</b>
Vyloučení úryvku: /drop .....	46
Přehled aktivních úryvků: /list .....	46
Přehled všech úryvků: /list -all .....	47
Přehled objektů daného druhu .....	47
Uložení aktivních úryvků: /save <file> .....	48
Uložení všech zadaných úryvků: /save -all <file> .....	49
Uložení dosavadního průběhu seance: /save -history <file> .....	49
Načtení skriptu: /open <file> .....	49
Ukončení seance: /exit .....	49
Restart: /reset .....	50
Znovuzavedení: /reload -restore .....	50
Nastavení startovního skriptu: /set -start <file> .....	50
Nastavení zpětnovazebního režimu: /set feedback .....	50
Aktivace dalšího zdroje: /env .....	51
Nápověda: /? .....	52
<b>1.7 Základní syntaktická pravidla .....</b>	<b>52</b>
Bílé znaky .....	52
Komentáře .....	53
<b>1.8 Ovládání .....</b>	<b>54</b>
Použití editoru .....	54
Nastavení vlastního editoru .....	57
<b>1.9 Doprovodné programy .....</b>	<b>57</b>
<b>1.10 Soubory pro opakování .....</b>	<b>57</b>
<b>2 Základní datové typy a jejich literály .....</b>	<b>58</b>
<b>2.1 Datové typy .....</b>	<b>58</b>
Dělení datových typů .....	59
Primitivní datové typy .....	60
Implicitní kódování znaků – UTF-8 .....	61
Objektové datové typy .....	61
Odkazy na objekty .....	62
<b>2.2 Literály .....</b>	<b>62</b>
Literály typu boolean .....	62
Literály typu int .....	63
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 exponentovém tvaru .....	67
Literály typu float .....	69
Literály typu char .....	69
Prázdný odkaz null .....	72
Literály typu String .....	73
Odbočka: volání metod .....	73
Příklady .....	74
Textové bloky .....	76
Literály typu Class .....	79
<b>2.3 Ještě trocha terminologie .....</b>	<b>79</b>
<b>2.4 Nestandardní hodnoty reálných typů .....</b>	<b>79</b>

2.5 Soubory pro opakování .....	81
<b>3 Proměnné .....</b>	<b>82</b>
3.1 Pravidla pro tvorbu identifikátorů .....	82
Používání znaku \$ .....	84
Konvence pro velikost písmen .....	84
3.2 Druhy typování .....	85
Statické × dynamické typování .....	85
Definice × odvození datového typu .....	85
Silné (přísné) × slabé typování .....	86
Shrnutí .....	86
3.3 Definice × deklarace .....	86
3.4 Deklarace a definice proměnných .....	87
3.5 Středníky .....	89
3.6 Současná deklarace více proměnných .....	90
Reakce prostředí <i>JShell</i> .....	90
3.7 Redeklarace proměnných v <i>JShell</i> .....	90
3.8 Deklarace s přiřazením počáteční hodnoty .....	92
Pozor na velikost znaků .....	92
Zpět k deklaraci s přiřazením počáteční hodnoty .....	93
3.9 Syntaktický diagram .....	94
3.10 Definice proměnných s využitím var .....	94
3.11 Soubory pro opakování .....	95
<b>4 Základní operátory .....</b>	<b>96</b>
4.0 Inicializace prostředí <i>JShell</i> .....	97
4.1 Nejprve trocha teorie .....	97
4.2 Operátor přiřazení = .....	98
Přiřazení je výraz .....	98
4.3 Unární + a - .....	99
4.4 Aritmetické operátory + - * / % .....	99
Operátor sčítání .....	99
Sčítání stringů .....	99
Operátor odčítání .....	100
Operátor násobení .....	101
Operátor dělení .....	101
Operátor zbytku po dělení .....	101
4.5 Kulaté závorky () .....	102
Alternativní řešení .....	103
4.6 Operátor přetypování ( <i>typ</i> ) .....	103
Implicitní přetypování .....	103
Příklady implicitního přetypování .....	104
Explicitní přetypování .....	106
Priorita .....	106
Kontrola .....	106
Explicitní přetypování hodnot primitivních typů .....	107
Příklady .....	107
Přetypování instancí objektových datových typů .....	109
Univerzální „přetypování“ na <i>String</i> .....	110
Textový podpis .....	111
4.7 Specifika číselných typů .....	111
Malé celočíselné typy .....	111
Ztráta přesnosti .....	113
Pořadí vyhodnocování .....	113
První příklad .....	114
Druhý příklad .....	114

4.8 Soubory pro opakování .....	114
<b>5 Definice metod.....</b>	<b>115</b>
5.1 Historické ohlédnutí.....	115
5.2 Definice a volání metody .....	116
5.3 Volání metody.....	118
5.4 Metody s parametry .....	119
Parametry versus argumenty .....	120
Více parametrů .....	121
Předávání hodnot parametrům .....	121
5.5 Metody vracující hodnotu .....	122
5.6 Přetěžování metod .....	122
5.7 Lokální proměnné metod.....	124
Postup volání metody .....	125
Parametry x lokální proměnné .....	125
Zásobník návratových adres – ZNA.....	126
Životnost lokálních proměnných .....	126
5.8 Příklady .....	127
Jídelna .....	128
Návratová hodnota .....	128
Definice metod v editoru.....	128
5.9 Metody s proměnným počtem argumentů.....	128
5.10 Přehled definovaných metod .....	129
5.11 Syntaktický diagram .....	130
5.12 Přímé spuštění souboru se sadou metod (preview).....	131
Definice spustitelného souboru.....	131
Překlad a spuštění vytvořeného programu.....	132
Omezení oproti JShell.....	134
5.13 Soubory pro opakování .....	134
<b>6 Ostatní operátory .....</b>	<b>135</b>
6.1 Inkrementační a dekrementační operátory ++ -- .....	135
6.2 Porovnávací operátory < <= == != >= > .....	137
Testování shody reálných čísel .....	138
Zvláštnosti porovnávání stringů .....	139
p12 == false.....	140
p13 == true.....	140
p23 == false.....	141
Porovnávání objektů reprezentujících hodnotu .....	141
6.3 Logické operátory ! & &&      .....	141
6.4 Bitové operátory ~ &   ^ << >> >>> .....	143
6.5 Složené přiřazovací operátory Op=.....	147
Příklady využití přetypování .....	147
6.6 Ternární operátor :? – podmíněný výraz.....	148
Ještě jednou porovnávání reálných čísel .....	150
6.7 Přepínač – výraz switch .....	151
Pravidla.....	152
Příklad.....	152
6.8 Operátor instanceof.....	153
Rozšíření funkcionality operátoru instanceof .....	156
6.9 Zbývající operátory: new [] () . .....	157
Operátor new .....	157
Operátor . (tečka).....	158
Operátor [] .....	159
Operátor volání metody () .....	159

6.10	Priorita, asociativita a komutativita operátorů .....	159
	Priorita .....	159
	Asociativita .....	161
	Komutativita .....	161
6.11	Šablonový výraz (předběžné – preview) .....	161
6.12	Soubory pro opakování .....	164
<b>7</b>	<b>Pole</b> .....	<b>165</b>
7.1	Strukturovaný datový typ – kontejner – pole .....	165
7.2	Deklarace a inicializace polí .....	167
	Syntaxe zděděná od jazyků C/C++ .....	168
7.3	Přiřazení hodnoty poli a přetypování polí .....	169
7.4	Počet prvků pole .....	170
7.5	Práce s prvky pole .....	171
7.6	Vícerozměrná pole – pole polí .....	173
	Obdélníková pole .....	174
	Zubatá pole .....	174
	Inicializace dvourozměrného pole .....	176
	Inicializace vícerozměrného pole .....	177
7.7	Proměnný počet argumentů metod .....	177
7.8	Arrays – knihovna metod pro práci s poli .....	178
7.9	Emulace předání argumentu odkazem .....	178
7.10	Pole a moderní programování .....	179
7.11	Soubory pro opakování .....	180
<b>8</b>	<b>Rozhodování</b> .....	<b>181</b>
8.1	Jednoduchý podmíněný příkaz .....	181
8.2	Blok příkazů (složený příkaz) .....	182
	Vnořování bloků .....	183
	Proměnné lokální v bloku .....	184
8.3	Rozpoznávání vzorů operátorem instanceof .....	187
	Lokální proměnná vytvořená v rámci vyhodnocování složitějšího výrazu .....	189
8.4	Úplný podmíněný příkaz .....	190
8.5	Složený podmíněný příkaz .....	191
8.6	Přepínač – příkaz switch .....	193
	Pravidla .....	194
	Příklad .....	196
	Rozšíření funkcionality .....	199
8.7	Soubory pro opakování .....	199
<b>9</b>	<b>Opakování části kódu</b> .....	<b>200</b>
9.1	Obecný cyklus .....	200
9.2	Cyklus s ukončovací podmínkou – cyklus do...while .....	201
9.3	Cyklus s počáteční podmínkou – cyklus while .....	202
9.4	Cyklus s parametrem – cyklus for .....	204
	Metody s proměnným počtem argumentů .....	206
9.5	„Dvojtečkový“ cyklus for (cyklus „for each“) .....	208
9.6	Vnořování cyklů .....	211
9.7	Cyklus s prázdným tělem .....	211
9.8	Nekonečný cyklus .....	212
9.9	Cyklus s podmínkou uprostřed .....	213
9.10	Příkaz break s návěstím .....	215
9.11	Příkaz continue .....	216
9.12	Rekurze .....	217
	Princip .....	218

Přímá a nepřímá rekurze .....	219
Přeplnění zásobníku návratových adres .....	219
9.13 Soubory pro opakování .....	220

## Část II Základní objektové konstrukce 221

<b>10 Základy objektově orientovaného paradigmatu .....</b>	<b>222</b>
10.1 Předmluva .....	222
Terminologická vsuvka .....	223
Syntaxe .....	223
Sémantika .....	223
Paradigma .....	223
Přehled paradigmat podporovaných <i>Javou</i> .....	224
Procedurální paradigma .....	224
Funkcionální paradigma .....	224
Objektově orientované paradigma .....	225
10.2 Trocha historie .....	225
10.3 Motivace OOP .....	226
10.4 Objekty .....	226
Členy objektů .....	227
10.5 Třídy a jejich instance .....	227
10.6 Třída jako objekt .....	228
10.7 Členy třídy a jejich instancí .....	229
Přežívající lokální proměnné .....	230
10.8 Zprávy .....	230
10.9 Metody .....	231
10.10 Entity .....	231
10.11 Polymorfismus, rozhraní, interfejs .....	232
Rozhraní × implementace .....	232
Atributy × vlastnosti .....	233
Vlastnosti v knihovně/platformě/frameworku <i>JavaFX</i> .....	234
Signatura × kontrakt .....	234
Rozhraní × interface .....	234
Interfejs a jeho instance .....	235
10.12 Objektové datové typy .....	236
10.13 Dědění .....	236
Jak potomek „rozšiřuje“ předka .....	237
Tři druhy dědění .....	237
Přirozené (nativní) dědění .....	238
Dědění typu (rozhraní) .....	238
Dědění implementace .....	239
Dva způsoby dědění, kachní typování .....	239
Statické a dynamické typování .....	239
Strukturální dědění .....	240
Kachní typování .....	240
Hierarchie jmenovitého dědění .....	240
Problémy s děděním – substituční princip Liskové (LSP) .....	240
10.14 Vlastní instance třídy a mateřská třída objektu .....	241
10.15 Tři základní principy OOP .....	242
10.16 Jazyk UML .....	243
10.17 Správa paměti .....	244
10.18 Další informace .....	245
10.19 Soubory pro opakování .....	245
<b>11 Třídy a jejich členy .....</b>	<b>246</b>
11.1 Nejjednodušší definice třídy .....	246

11.2	Konstruktory.....	247
	Implicitní konstruktor.....	247
	Vlastní konstruktor a skrytý parametr <code>this</code> .....	247
	Proč se liší podpisy.....	248
	Definice tříd jako úryvky.....	249
11.3	Třída se všemi členy.....	249
	Statické (třídní) členy.....	250
	Instanční členy.....	251
	Konstrukce objektů.....	252
11.4	Kvalifikace posílaných zpráv.....	252
	Implicitní kvalifikace.....	253
11.5	Přetěžování konstruktorů.....	254
	Kvalifikace klíčovým slovem <code>this</code> .....	257
11.6	Modifikátory přístupu a skrývání implementace.....	258
	Veřejné a „neveřejné“ datové typy.....	259
11.7	Přístupové metody.....	259
11.8	Modifikátor <code>final</code> .....	261
	Konstantní atributy.....	261
	Konstanty vyhodnotitelné v době překladu.....	261
	Konstantní lokální proměnné.....	262
	Efektivní konstanty.....	262
	Zveřejňování konstantních atributů.....	262
	Modifikátor <code>final</code> v procesu dědění.....	262
	Neměnnost objektů.....	263
11.9	Primitivní a obalové datové typy – autoboxing.....	263
	Převody stringů na hodnoty primitivních typů.....	265
11.10	Důležité metody klíčových tříd.....	266
	Třída <code>Object</code> .....	266
	<code>Object clone()</code> .....	266
	Mělké a hluboké kopie objektů.....	266
	<code>boolean equals(Object)</code> .....	266
	<code>Class&lt;?&gt; getClass()</code> .....	267
	<code>int hashCode()</code> .....	267
	<code>String toString()</code> .....	267
	<code>void finalize()</code> .....	267
	Třída <code>String</code> .....	268
	<code>char charAt(int index)</code> .....	268
	<code>boolean contains(CharSequence s)</code> .....	268
	<code>boolean equals(Object)</code> .....	268
	<code>boolean equalsIgnoreCase(String anotherString)</code> .....	268
	<code>int indexOf(int ch)</code> .....	268
	<code>int indexOf(String ch)</code> .....	268
	<code>boolean isEmpty()</code> .....	268
	<code>static String join(CharSequence delimiter, CharSequence... elements)</code> <code>static String join(CharSequence delimiter, Iterable&lt; extends CharSequence&gt; elements)</code> .....	268
	<code>int length()</code> .....	268
	<code>String replace(X oldX, X newX)</code> .....	268
	<code>String[] split(String regex)</code> .....	268
	<code>String strip()</code> <code>String trim()</code> .....	269
	<code>String toLowerCase()</code> <code>String toUpperCase()</code> .....	269
	<code>static String valueOf(X x)</code> .....	269
	Třída <code>Class</code> .....	269
	<code>boolean equals(Object)</code> .....	269
	<code>String getName()</code> .....	269

String <code>getSimpleName()</code> .....	269
String <code>toString()</code> .....	269
11.11 Soubory pro opakování .....	269
<b>12 Vytvoření aplikace a vývojová prostředí .....</b>	<b>270</b>
12.1 Doprovodné programy .....	271
12.2 Malé ohlédnutí .....	271
12.3 Zdrojové soubory třídy .....	272
12.4 Představení balíčků .....	274
Spuštění třídy z balíčku .....	275
12.5 Překlad .....	276
Překlad třídy .....	276
Překlad rozsáhlejšího programu .....	278
12.6 Používání knihoven .....	279
12.7 JAR-soubory .....	279
Vytvoření JAR-souboru .....	280
Vytvoření knihovny .....	280
Vytvoření spustitelného JAR-souboru .....	282
12.8 Spuštění JAR-souborů .....	283
Syntaktický diagram spuštění aplikace .....	284
Java .....	284
ArgumentVM .....	284
Spouštěná třída .....	284
ArgumentProgramu .....	285
12.9 Spustitelný program používající knihovnu .....	285
12.10 Vývojová prostředí – IDE .....	287
JShell .....	287
<i>BlueJ</i> a <i>BlueJ++</i> .....	288
Nejpoužívanější profesionální IDE .....	288
IntelliJ IDEA .....	288
Eclipse .....	289
NetBeans .....	289
Visual Studio Code .....	289
12.11 Spuštění hlavní třídy a její metody <code>main(String[])</code> .....	290
12.12 Soubory pro opakování .....	290
<b>13 Balíčky a knihovny .....</b>	<b>291</b>
13.1 Velké programy a jejich problémy .....	291
13.2 Balíčky .....	292
Umístění zdrojových souborů .....	293
Kořenový (implicitní, defaultní, nepojmenovaný) balíček .....	293
Podbalíčky .....	294
Konvence pro názvy balíčků .....	294
Balíčky doprovodných programů a knihoven .....	294
Zakázaný balíček java .....	295
Názvy datových typů .....	296
13.3 Explicitní ukončení aplikace .....	297
13.4 Příkaz <code>import</code> .....	298
Import zadaného datového typu .....	298
Import všech veřejných typů ze zadaného balíčku .....	298
Podpora zadávání příkazu <code>import</code> ve vývojových prostředích .....	299
Příkaz <code>import</code> v prostředí JShell .....	300
Výjimečnost balíčku <code>java.lang</code> .....	300
13.5 Příkaz <code>import static</code> .....	300
13.6 Syntaktický diagram příkazu <code>import</code> .....	301
13.7 Entity soukromé v rámci balíčku .....	302

13.8	Typy se stejným názvem v různých balíčcích .....	304
	Shrnutí .....	306
13.9	Použití knihovny v JShell .....	306
	Nastavení proměnné <code>classpath</code> .....	306
	Nastavení importů .....	306
	Ukázky .....	307
	Násilné ukončení aplikace .....	307
13.10	Zakázaný balíček <code>java</code> .....	309
13.11	Soubory pro opakování .....	309
<b>14</b>	<b>Dokumentace API .....</b>	<b>310</b>
14.1	Dokumentační komentáře a API .....	310
14.2	Proč psát srozumitelné a komentované programy .....	311
	POBLIOCHA .....	312
	Jak dokumentační komentáře zobrazovat .....	313
14.3	Jak psát dokumentační komentáře pro <code>javadoc</code> .....	313
14.4	Pomocné značky pro tvorbu dokumentace .....	314
14.5	Dokumentace balíčku a modulu .....	315
14.6	Vytvoření a zobrazení dokumentace .....	317
14.7	Struktura dokumentace API .....	317
	Struktura dokumentace datového typu .....	318
	Rychlé vyhledání .....	319
14.8	Zpřehlednění programu .....	319
14.9	Zakomentování a odkomentování části programu .....	321
14.10	Soubory pro opakování .....	321
<b>15</b>	<b>Konstrukce interface .....</b>	<b>322</b>
15.1	Definice typického interfejsu .....	322
	Deklarace abstraktních metod .....	323
	Příklad .....	323
15.2	Implementace interfejsu třídou .....	324
15.3	Interfejs se všemi přípustnými typy členů .....	326
	Motivace k rozšíření – implicitní metody .....	326
	Statické členy .....	328
	Instanční členy .....	328
15.4	Dědění interfejsů .....	329
15.5	Příklad .....	329
	Plynulé posuny .....	330
	Plynulé změny velikosti .....	331
	Sloučení knihoven .....	331
15.6	Výhody implicitních metod při návrhu architektury .....	332
15.7	Řešení kolizí .....	333
15.8	Specifikace zdroje použité metody .....	335
	Možné problémy .....	335
15.9	Speciální interfejsy .....	337
	Značkovací interfejsy .....	337
	<code>java.lang.Cloneable</code> .....	337
	<code>java.io.Serializable</code> .....	337
	Současné trendy a doporučení .....	337
	Funkční interfejsy .....	337
	Interfejs <code>Iterable</code> .....	338
15.10	Soubory pro opakování .....	339
<b>16</b>	<b>Podrobnosti o konstruktorech .....</b>	<b>340</b>
16.1	Opakování: co víme o konstruktorech instancí .....	340
16.2	Zavádění třídy – <code>java.lang.ClassLoader</code> .....	341



16.3	Statický konstruktor – konstruktor třídy .....	342
	Konstruktor interfejsu .....	342
16.4	Instanční inicializační blok .....	343
16.5	Dvě části těla konstruktoru instancí.....	343
16.6	Příklad .....	344
	Konstruktor třídy .....	349
	3-9: Úvodní statický inicializační blok .....	349
	25: Předčasné použití atributu .....	350
	8: Nekorektní použití metod .....	350
	42: Předčasné použití konstanty .....	350
	62: Nekorektní volání konstruktoru .....	351
	Inicializační část konstruktoru instancí .....	351
	12-15: Úvodní instanční inicializační blok .....	351
	149: Deklarace konstanty <code>loaded</code> .....	351
	151-155: Inicializační výpočet .....	352
	163: Použití <code>this</code> v inicializaci .....	352
	260-264: Závěrečný inicializační blok .....	352
	Těla konstruktorů instancí .....	352
	175-180: Bezparametrický konstruktor .....	352
	188-194: Jednparametrický konstruktor .....	353
	203-208: Dvoupametrický konstruktor .....	353
	218-231: Tříparametrický konstruktor .....	353
16.7	Experimenty .....	353
16.8	Doporučení .....	354
	Jediný statický inicializační blok .....	354
	Bez instančních inicializačních bloků .....	354
	Inicializovat všechny atributy jednotně.....	355
16.9	Skutečný název metody konstruktoru .....	355
16.10	Soubory pro opakování .....	356
17	Úvod do dědění implementace.....	357
17.1	Úvodní poznámky.....	357
17.2	Definice dceřině třídy.....	358
17.3	Rodičovský podobjekt .....	360
	Dědění implementace od více rodičů .....	361
17.4	Konstruktor .....	361
	Konstrukce rodičovského podobjektu .....	362
17.5	Přetížené verze konstruktorů, použití <code>super</code> × <code>this</code> .....	364
17.6	Konstruktory rodiče a potomka .....	365
17.7	Demonstrace chování konstruktorů.....	366
	Definice třídy <code>Graddaughter17</code> .....	366
	Provedení akce před příkazem <code>this()</code> nebo <code>super()</code> .....	368
	Definice metody <code>constructorReport(Object, Class)</code> .....	369
	Spuštění testu .....	370
	Zavedení třídy .....	370
	Tisk nehotových objektů .....	370
	Preference vlastních metod .....	372
	Dokončení testu .....	372
	Rodičovský podobjekt je abstrakce .....	372
17.8	Dědění přístupových práv .....	372
17.9	Zákaz vytváření potomků třídy.....	373
17.10	Zalepené třídy a interfejsy .....	373
	Definice potomků uvnitř definice zalepeného typu.....	374
	Definice potomků uvnitř definice zalepeného typu.....	374
	Pravidla pro potomky .....	376
17.11	Soubory pro opakování .....	379

<b>18 Viditelnost členů tříd</b> .....	<b>380</b>
18.1 Úpravy použitého kódu.....	380
18.2 Trocha terminologie.....	381
Posílání zpráv a volání metod .....	381
Přetěžování×přebíjení×zakrývání×přepisování×předefinování metod.....	381
Přetěžování metod.....	381
Přebíjení metod .....	381
Zakrývání metod .....	382
Přepsání či předefinování metod.....	382
18.3 Chráněné členy – modifikátor přístupu <code>protected</code> .....	382
Shrnutí .....	385
18.4 Dědění metod.....	385
Zděděné, dále neupravované metody .....	386
Zděděné metody, pro něž potomek definuje „lepší“ implementaci .....	386
Kompatibilita signatur.....	386
18.5 Zakrývání metod předka (method hiding).....	387
18.6 Metody, které není možno v potomku zakrýt či přebít – modifikátor <code>final</code> .....	390
18.7 Zakrývání atributů předka .....	391
18.8 Metody nově definované v potomku .....	393
Proč je situace jednoduchá jen zdánlivě.....	393
Anotace <code>@Override</code> .....	393
Statically × dynamicky typované jazyky.....	394
18.9 Závěr.....	395
18.10 Soubory pro opakování .....	395
<b>19 Virtuální metody a jejich přebíjení</b> .....	<b>396</b>
19.1 Princip .....	396
Časná a pozdní vazba .....	397
Virtuální metody.....	397
19.2 Které metody jsou v Javě virtuální .....	398
19.3 Chování virtuálních metod .....	398
19.4 Zdokonalení třídy <code>Square</code> – třída <code>Square19</code> .....	401
Přebíjení metody <code>copy()</code> .....	401
Problémy s nastavováním velikosti.....	401
První návrh definice metody <code>setSize(int, int)</code> .....	402
Test prvního návrhu .....	403
Oprava .....	404
19.5 Co se nám na dědění nelíbí.....	406
19.6 Soubory pro opakování .....	406
<b>20 Abstraktní třídy</b> .....	<b>407</b>
20.1 Abstraktní třídy a jejich role v dědické hierarchii .....	407
Vytváříme hybrida.....	408
Abstraktní třída bez abstraktních metod.....	409
20.2 Konstruktor abstraktní třídy.....	409
20.3 Deklarace a implementace abstraktních metod .....	410
20.4 Účel abstraktních tříd.....	412
20.5 Proč společný rodič.....	413
20.6 Účel abstraktních metod.....	413
20.7 Návrhový vzor Šablonová metoda (Template method) .....	414
Princip.....	414
Implicitní metody interfejsů.....	414
Architektura balíčku <code>ru1ib.geom</code> .....	415
Metoda <code>toString()</code> .....	416
20.8 Soubory pro opakování .....	418

<b>Část III Pokročilejší objektové konstrukce</b>	<b>419</b>
<b>21 Výjimky a aserce</b>	<b>420</b>
21.1 Co to jsou výjimky	421
21.2 Analýza chybové zprávy	421
Oznámení o chybě	421
Jak chyba vznikla – výpis zásobníku návratových adres	422
21.3 Nejdůležitější výjimky	423
21.4 Vyhození výjimky	424
Oddělené vytvoření výjimky	426
21.5 Výjimky a nedosažitelný kód	427
21.6 Co výjimky umí	427
21.7 Hierarchie dědění výjimek	428
21.8 Zachycení vyhozené výjimky	429
Chování metody <code>exceptionCatching(int)</code>	431
21.9 Syntaktický diagram bloku <code>try ... catch</code>	431
Několik současně odchyťovaných výjimek	431
Společná reakce na několik výjimek	432
Společný úklid – blok <code>finally</code>	433
Příklad	433
21.10 Definice vlastních výjimek	435
21.11 Kontrolované výjimky	436
21.12 Převedení kontrolované výjimky na nekontrolovanou	438
21.13 Informace o skutečném původci výjimky	439
21.14 Ověřování podmínek – příkaz <code>assert</code>	441
Design by Contract	442
Výběrová aktivace provádění příkazu <code>assert</code>	444
21.15 Kdopak mne to volal	446
21.16 Soubory pro opakování	446
<b>22 Generické datové typy a metody</b>	<b>447</b>
22.1 Motivace	447
22.2 Generické a parametrizované datové typy	450
22.3 Definice generických typů	451
22.4 Použití generických typů	453
22.5 Překlad generických datových typů a očišťování	455
22.6 Rizika nepoužití typových argumentů	456
22.7 Varování překladače a jejich potlačení	459
Zobrazená varování	460
22.8 Generické metody	461
22.9 Soubory pro opakování	465
<b>23 Typové parametry a argumenty</b>	<b>466</b>
23.1 Omezení typových argumentů	466
Typové argumenty s více předky	467
Vzájemné závislosti typových parametrů	467
23.2 Překlad a očišťování podrobněji	468
Doporučené pořadí omezujících interfejsů	468
Ztráta informace při běhu	470
Přemosťovací metody	470
23.3 Zakázané operace	472
Za typové parametry nelze dosazovat primitivní typy	472
Typové parametry třídy není možno použít u statických členů	472
Nelze vytvořit instanci typového parametru	472
Reflexe	473

Nelze vytvořit pole instancí typového parametru ani parametrizovaného typu .....	474
Výjimky .....	475
23.4 Proměnný počet argumentů – @SafeVarargs .....	475
Omezení .....	476
Vytvoření pole hodnot .....	477
23.5 Nejednoznačnosti a kolize .....	477
Falešně přetížená metoda .....	477
Nová metoda koliduje se zděděnou .....	478
Kolize požadovaných interfejsů .....	479
Kolize implementovaných interfejsů .....	480
Potomci a předci generických typů – špatné pochopení dědičnosti .....	480
23.6 Žolíky .....	481
23.7 Příklad: datový typ <code>Interval&lt;T extends Comparable&lt;? super T&gt;&gt;</code> .....	483
23.8 Soubory pro opakování .....	486
<b>24 Interní datové typy .....</b>	<b>487</b>
24.1 Motivace .....	487
Pomocný soukromý typ .....	487
Objekt znající útroby a implementující veřejné rozhraní .....	488
Sdružení souvisejících typů .....	488
24.2 Terminologie .....	488
24.3 Společné charakteristiky interních typů .....	490
24.4 Globální interní (členské) datové typy .....	490
24.5 Vnořené datové typy .....	491
24.6 Vnitřní třídy .....	492
Interní interfejsy a výčtové typy bez modifikátoru <code>static</code> .....	492
24.7 Příklad na vnořené a vnitřní třídy .....	493
Vnořená <code>Elements</code> × vnitřní <code>SAIterator</code> .....	494
Veřejná <code>Elements</code> × soukromá <code>SAIterator</code> .....	494
Definice třídy <code>SparseArray.Element</code> .....	495
Definice třídy <code>SparseArray.SAIterator</code> .....	495
Definice třídy <code>SparseArrayTest</code> .....	496
24.8 Lokální třídy a interfejsy .....	498
Pojmenované lokální třídy .....	499
Anonymní třídy .....	499
Použití anonymních tříd .....	501
24.9 Ještě jednou názvy tříd .....	501
24.10 Soubory pro opakování .....	502
<b>25 Výčtové typy – třídy typu <code>enum</code> .....</b>	<b>503</b>
25.1 Nejjednodušší definice .....	503
Překladačem přidané atributy a metody .....	505
Atribut <code>\$VALUES</code> .....	505
<code>public static final NázevTypu[] values()</code> .....	506
<code>public static NázevTypu valueOf(String name)</code> .....	506
25.2 Třída <code>Enum</code> .....	506
Zděděné metody .....	510
<code>public final String name()</code> .....	510
<code>public final int ordinal()</code> .....	510
<code>public final int compareTo(E o)</code> .....	510
<code>public final Class&lt;E&gt; getDeclaringClass()</code> .....	510
<code>public static &lt;T extends Enum&lt;T&gt;&gt; T valueOf(Class&lt;T&gt; enumType, String name)</code> .....	511
Přebité verze metod zděděných od třídy <code>Object</code> .....	511
<code>protected final Object clone() throws CloneNotSupportedException</code> .....	511
<code>equals(Object)</code> .....	511