

knihovna programátora

- Podrobný výklad vlastností jazyka od naprostých základů až po pokročilé, běžně neprobírané konstrukce
- Probírá i konstrukce, které budou zařazeny až do příštích verzí jazyka, a předvádí, jak tyto konstrukce vyzkoušet
- 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



RUDOLF PECINOVSKÝ

Java 14

Kompletní příručka jazyka

Ing. Rudolf Pecinovský, CSc. je absolventem *Fakulty Elektrotechnické ČVUT* z roku 1979. Titul CSc. získal v *Ústavu teorie informace a automatizace ČSAV* v roce 1983. Od počátku 80. let učí a publikuje, přičemž svůj výzkum soustředí především na oblast vstupních kurzů moderního programování pro naprosté začátečníky. V současné době učí na *Vysoké škole ekonomické v Praze*, na *Fakultě jaderné a fyzikálně inženýrské ČVUT* a na *Vysoké škole podnikání a práva*. Doposud mu vyšlo přes 60 knih, které byly přeloženy do pěti jazyků. Většina jeho knih je zaměřena na výuku moderního programování a na umění návrhu objektově orientované architektury.



knihovna programátora

RUDOLF PECINOVSKÝ

Java 14

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 14

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 7620. publikaci

Odpovědný redaktor: Ivana Palasová

Návrh vnitřního layoutu Rudolf Pecinovský

Zlom Rudolf Pecinovský

Počet stran 578

První vydání, Praha 2020

Vytiskly Tiskárny Havlíčkův Brod, a.s.

© Grada Publishing, a.s., 2020

Cover Design © Grada Publishing, a. s., 2020

Cover Photo © Depositphotos

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

ISBN 978-80-271-1605-8 (ePub)

ISBN 978-80-271-1604-1 (pdf)

ISBN 978-80-271-1369-9 (print)

Všem, kteří se chtějí něco naučit

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 <code>interface</code>	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	416
23 Typové parametry a argumenty	436
24 Interní datové typy	457
25 Výčtové typy – třídy typu enum	472
26 Záznamy – třídy typu record.....	486
27 Lambda-výrazy	500
28 Anotace	510
29 Vlákna a paralelní procesy	523
30 Moduly.....	531
Část IV Přílohy	555
A Omezení JVM	556
B Tvorba jednoduchého GUI.....	557
Literatura.....	571
Rejstřík.....	573

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 JShell	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 JShell	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>JShell</i>	33
1.3 Problémy s klávesnicí	33
1.4 Příprava programu <i>JShell</i> 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: <code>/drop</code>	40
Přehled aktivních úryvků: <code>/list</code>	40

Přehled všech úryvků: <code>/list -all</code>	41
Přehled objektů daného druhu	42
Uložení aktivních úryvků: <code>/save <file></code>	43
Uložení všech zadaných úryvků: <code>/save -all <file></code>	43
Uložení dosavadního průběhu seance: <code>/save -history <file></code>	43
Načtení skriptu: <code>/open <file></code>	43
Ukončení seance: <code>/exit</code>	43
Restart: <code>/reset</code>	44
Znovuzavedení: <code>/reload -restore</code>	44
Nastavení startovního skriptu: <code>/set -start <file></code>	44
Nápověda: <code>/?</code>	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 <code>boolean</code>	55
Literály typu <code>int</code>	55
Historická vsuvka – číselné soustavy	56
Názvy skupin bitů	57
Literály typu <code>long</code>	57
Literály typu <code>byte</code> a <code>short</code>	58
Literály typu <code>double</code>	58
Celé číslo s příponou	59
Obyčejné desetinné číslo	59
Číslo v exponentovém tvaru	59
Literály typu <code>float</code>	61
Literály typu <code>char</code>	61
Prázdný odkaz <code>null</code>	65
Literály typu <code>String</code>	65
Textové bloky	67
Literály typu <code>Class</code>	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 <code>\$</code>	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.....	82
	Pozor na velikost znaků.....	83
	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 trocha 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 (typ).....	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 polí 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 <code>switch</code>	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 <code>do...while</code>	181
9.3 Cyklus s počáteční podmínkou – cyklus <code>while</code>	182
9.4 Cyklus s parametrem – cyklus <code>for</code>	184
Metody s proměnným počtem argumentů.....	186
9.5 „Dvojtečkový“ cyklus <code>for</code> (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 <code>break</code> s návěštím.....	194
9.11 Příkaz <code>continue</code>	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 × implementace	210
	Atributy × vlastnosti	211
	Vlastnosti v knihovně/platformě/frameworku <i>JavaFX</i>	211
	Signatura × kontrakt	212
	Rozhraní × 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 this	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 this	232
11.6	Modifikátory přístupu a skrývání implementace	233
	Veřejné a „neveřejné“ datové typy	234
11.7	Přístupové metody	234
11.8	Modifikátor final	235
	Konstantní atributy	236
	Konstanty vyhodnotitelné v době překladu	236
	Konstantní lokální proměnné	236
	Efektivní konstanty	237
	Zveřejňování konstantních atributů	237
	Modifikátor final 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 Object	240
	Object clone()	240
	Mělké a hluboké kopie objektů	240
	boolean equals(Object)	241
	Class<?> getClass()	241
	int hashCode()	241
	String toString()	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 NetBeans	245
12.3 Vytvoření spustitelného projektu v NetBeans	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 spuštění 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 NetBeans	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 import static	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
	POBLIOCHA	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 protected	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 (method hiding).....	355
18.6 Metody, které není možno v potomku zakrýt či přebít – modifikátor final	357
18.7 Zakrývání atributů předka	359
18.8 Metody nově definované v potomku	360
Proč je situace jednoduchá jen zdánlivě.....	361
Anotace @Override	361
Statically × 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 Square	368
Přebíjení metody copy()	369
Problémy s nastavováním velikosti.....	369
První návrh definice metody setSize(int, int)	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 (Template method)	381
Princip.....	381
Implicitní metody interfejsů.....	382
Architektura balíčku eu.pedu.lib20s.geom	382
Metoda toString()	384
20.8 Shrnutí.....	386