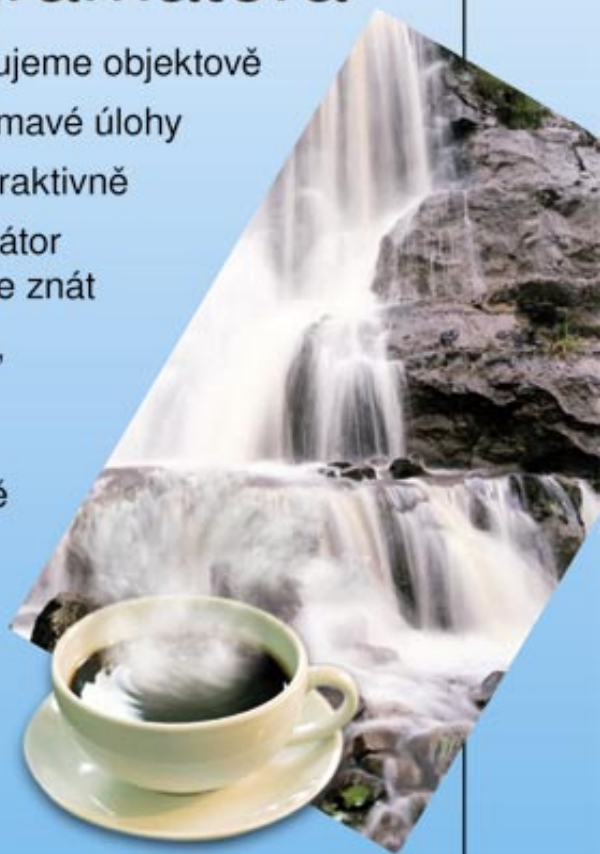


# knihovna programátora

- Od samého počátku programujeme objektivě
- Již od prvních kapitol řeší zajímavé úlohy
- Ukazuje, jak programovat interaktivně
- Probírá oblasti, které programátor ve své praxi opravdu potřebuje znát
- Věnuje se návrhovým vzorům, refaktorování, programování řízenému testy
- Doporučená učebnice na řadě středních a vysokých škol

myslíme  
objektivě  
v jazyku



# Java

RUDOLF PECINOVSKÝ

kompletní učebnice pro začátečníky, 2., aktualizované a rozšířené vydání

## 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**.

*Používání elektronické verze knihy je umožněno jen osobě, která ji legálně nabyla a jen pro její osobní a vnitřní potřeby v rozsahu stanoveném autorským zákonem. Elektronická kniha je datový soubor, který lze užívat pouze v takové formě, v jaké jej lze stáhnout s portálu. Jakékoliv neoprávněné užití elektronické knihy nebo její části, spočívající např. v kopírování, úpravách, prodeji, pronajímání, půjčování, sdělování veřejnosti nebo jakémkoliv druhu obchodování nebo neobchodního šíření je zakázáno! Zejména je zakázána jakákoliv konverze datového souboru nebo extrakce části nebo celého textu, umístování textu na servery, ze kterých je možno tento soubor dále stahovat, přitom není rozhodující, kdo takovéto sdílení umožnil. Je zakázáno sdělování údajů o uživatelském účtu jiným osobám, zasahování do technických prostředků, které chrání elektronickou knihu, případně omezují rozsah jejího užití. Uživatel také není oprávněn jakkoliv testovat, zkoušet či obcházet technické zabezpečení elektronické knihy.*





Copyright © Grada Publishing, a.s.

## O autorovi

Rudolf Pecinovský patří ke špičkovým odborníkům na výuku programování. Publikoval již 39 učebnic, které byly přeloženy do pěti jazyků, a nepřeberné množství článků a příspěvků na odborných konferencích. Je autorem metodiky výuky programování *Karel*, navazující metodiky *Baltík* a moderní metodiky výuky objektově orientovaného programování známé pod anglickým názvem *Design Patterns First*. Učí programování na VŠE a současně pracuje jako Senior EDU Expert ve firmě ICZ a.s., kde má na starosti doškolování profesionálních programátorů.



## O knize

Tato kniha je druhým vydáním populární učebnice programování, která je na našem trhu zcela ojedinělá. Na rozdíl od ostatních učebnic se totiž neomezuje na výuku syntaxe jazyka a práce s knihovnamy, ale učí čtenáře doopravdy programovat. Učí jej, jak má při programování myslet.

Kniha je sice primárně určena začátečníkům, ale ohlasy na první vydání ukázaly, že v ní najdou poučení i zkušení programátoři. Většina učebnic a kurzů programování totiž vyvolává falešnou představu, že objektově programovat znamená používat třídy a dědičnost. Tato kniha je první, která ukazuje, že objektově orientované programování přináší především jiný způsob myšlení. Jak výstižně napsal jeden čtenář: „*Myslel jsem si, že nejsem žádný programátorské ucho. Když jsem ale přečetl vaši učebnici, otevřel jsem oči a hubu. Konečně jsem pochopil věci, které mi ostatní učebnice nedokázaly vysvětlit.*“

Kniha vznikla na základě dlouholetých autorových zkušeností se školením profesionálních programátorů, výukou programování na univerzitě i vedením žákovských programátorských kroužků. Autor v ní uvádí čtenáře krok za krokem do tajů objektově orientovaného programování a ukazuje mu, jak možnosti této moderní technologie co nejlépe využít a kde si dát naopak pozor na její úskalí.

Výklad je postaven na příkladech, které autor spolu s čtením postupně řeší a přitom čtenáře učí nejenom základním programátorským návykům a dovednostem, ale předvede mu i nejrůznější užitečné triky, z nichž mnohé nikde jinde vysvětlené nenajdete. Současně upozorňuje na nejčastější začátečnické chyby, které před svými čtenáři ostatní učebnice většinou tají. Navíc probírá i řadu témat (např. návrhové vzory), které patří do základní výbavy objektového programátora, přestože jsou většinou probírána až v pokročilých nebo dokonce nadstavbových kurzech.

## Kurzy,

kteří vede Rudolf Pecinovský, patří k nejkvalitnějším v České republice a zaručují dokonalé pochopení problematiky a okamžitou využitelnost získaných vědomostí v praxi. Je v nich optimálně sklouben výklad principů s praktickými cvičeními a nácvikem samostatného řešení praktických úloh.

**Chcete-li se naučit opravdu dobře programovat,** přihlaste se do některého z následujících kurzů:

- ☞ **Úvod do objektově orientovaného programování pro neprogramátory** je určen pro ty, kteří se nehodlají žít přímo programováním, ale musejí s programátory velmi často jednat a potřebují se v dané oblasti trochu vyznat. Navštěvují jej zejména analytici, vedoucí projektových týmů a manažeři.
- ☞ **Úvod do objektově orientovaného programování v Javě pro začínající programátory** je určen pro ty, kteří s programováním teprve začínají a nemají žádné (a nebo jen minimální) předchozí zkušenosti s programováním.
- ☞ **Úvod do objektově orientovaného programování v Javě pro „strukturované“ programátory** je určen pro ty, kteří doposud programovali v některém strukturovaném jazyce, a nebo programovali v objektovém jazyce, ale cítí, že jim objektově orientovaný způsob myšlení není vlastní. Kurz navštěvují především programátoři, kteří začali programovat v PHP nebo Delphi a přecházejí na Javu. Neméně početnou skupinou jsou programátoři v Javě, kteří se v předchozích kurzech sice naučili syntaxi jazyka, ale cítí, že by potřebovali zlepšit „objektovou orientovanost“ svých programů.
- ☞ **Kurz programování v Javě pro pokročilé** je určen pro posluchače se základními zkušenostmi s objektovým programováním a Javou. Prohlubuje jejich znalosti a soustředí se na oblasti, které základní kurzy většinou přeskakují nebo je probírají jen okrajově. Posluchači se naučí pracovat s mnoha užitečnými třídami ze standardní knihovny a osvojí si řadu pokročilých technologií.
- ☞ **Kurz návrhových vzorů** je určen pro programátory se základními znalostmi objektového programování. Seznámí se zde s 33 návrhovými vzory a naučí se je využívat ve svých programech.

**Vedle těchto standardních kurzů nabízíme i další odborné akce:**

- ☞ **Přednášky a série přednášek** na domluvená témata pro větší skupiny posluchačů. Tyto přednášky mohou mít i podobu klasického výukového kurzu.
- ☞ **Konzultace** nad konkrétními problémy zákazníků.

Podrobnější informace najdete na [www.amaio.cz](http://www.amaio.cz)  
Dotazy a přihlášky můžete posílat na [kurzy@amaio.cz](mailto:kurzy@amaio.cz)



**Rudolf Pecinovský**

# **Myslíme objektově v jazyku Java**

kompletní učebnice pro začátečníky, 2., aktualizované a rozšířené vydání

Copyright © Grada Publishing a.s., 2009

V knize použité názvy mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

Vydala Grada Publishing a.s.

U Průhonu 22, Praha 7

jako svoji 3472. publikaci

Odborní lektori Prof. RNDr. PhDr. Antonín Slabý, CSc.,

Doc. Ing. Vojtěch Merunka, Ph.D., Ing. Alena Buchalceková, Ph.D.

Odpovědná redaktorka Jaroslava Palasová

Návrh vnitřního layoutu Rudolf Pecinovský

Zlom Jana Davidková, Rudolf Pecinovský

Počet stran 576

První vydání, Praha 2009

Vytiskly Tiskárny Havlíčkův Brod, a.s.

Husova Ulice 1881, Havlíčkův Brod

ISBN 978-80-247-2653-3 (tištěná verze)

ISBN 978-80-247-7025-3 (elektronická verze ve formátu PDF)

© Grada Publishing, a.s. 2011

*Mé ženě Jarušce a dětem  
Štěpánce, Pavlínce, Ivance a Michalovi*

# Stručný obsah

Poděkování .....	21
Předmluva k prvnímu vydání.....	22
Úvod .....	23
<b>Část 1: Zapouzdření</b> .....	<b>33</b>
1. Seznamujeme se s nástroji .....	34
2. Třídy a objekty v interaktivním režimu.....	51
3. Vytváříme vlastní třídu .....	85
4. Přidáváme atributy a metody.....	116
5. Dotváříme vlastní třídu.....	184
6. Návrhové vzory.....	229
<b>Část 2: Více tváří</b> .....	<b>243</b>
7. Rozhraní .....	244
8. Budete si to přát zabalit? .....	286
9. Co takhle něco zdědit?.....	307
10. Dědit mohou i třídy .....	324
11. Knihovny .....	389
12. Program ve výjimečné situaci .....	403
<b>Část 3: Učíme program přemýšlet</b> .....	<b>419</b>
13. Program začíná přemýšlet .....	420
14. Ještě jednu rundu, prosím.....	453
15. Interní datové typy .....	475
16. Kontejnery nejsou jen na odpadky .....	488
17. Statické kontejnery – pole .....	538
18. Závěrečný projekt a kudy dál.....	558
Rejstřík .....	565



# Podrobný obsah

Poděkování.....	21
Předmluva k prvnímu vydání .....	22
Úvod .....	23
Komu je kniha určena.....	23
Co se naučíte .....	23
Styl výuky .....	24
Programovací jazyk .....	25
Uspořádání .....	26
Čeština .....	26
Proč je kniha tlustá .....	27
Potřebné vybavení .....	27
Doporučená konfigurace.....	27
Sada JDK (Java Development Kit).....	28
Vývojové prostředí.....	28
Konfigurační soubor pro prostředí BlueJ .....	29
Doprovodné programy .....	29
Doprovodné animace.....	29
Použité konvence.....	29
Odbočka.....	31
<b>Část 1: Zapouzdření</b> .....	<b>33</b>
<b>1. Seznamujeme se s nástroji.....</b>	<b>34</b>
1.1 Trochu historie .....	34
První počítače.....	34
Co je to program .....	35
Program musí být především spolehlivý.....	35
1.2 Objektově orientované programování – OOP.....	36
Vývoj metodik programování.....	36
Principy OOP .....	37
1.3 Překladače, interprety, platformy .....	37
Operační systém a platforma.....	37
Programovací jazyky .....	38
1.4 Java a její zvláštnosti.....	39
Klíčové vlastnosti Javy.....	40
Objektově orientovaná .....	40
Jednoduchá .....	40
Multiplatformní.....	40
Java je jazyk i platforma .....	40
Vývojářská sada .....	41
1.5 Vývojové prostředí BlueJ .....	41
1.6 Projekty a BlueJ .....	42
Umístění projektů na disku .....	42
Windows a substituované disky.....	43
Vyhledání a otevření projektu .....	44

1.7	Diagram tříd.....	45
	Manipulace s třídami v diagramu .....	46
1.8	Shrnutí – co jsme se naučili.....	49
<b>2.</b>	<b>Třídy a objekty v interaktivním režimu.....</b>	<b>51</b>
2.1	Nejprve trochu teorie .....	51
	Třídy a jejich instance .....	51
	Zprávy .....	52
	Metody .....	52
2.2	Analogie.....	53
2.3	Třídy a jejich instance .....	53
	Vytváříme svou první instanci.....	53
	Pravidla pro tvorbu identifikátorů v jazyce Java .....	56
	Vytváříme svou první instanci – pokračování.....	57
	Posíláme instanci zprávu.....	59
	Vytváříme další instance .....	59
	Rušení instancí a správa paměti .....	60
2.4	Restartování virtuálního stroje .....	61
2.5	Instance versus odkaz.....	61
2.6	Zprávy žádající o hodnotu.....	63
	Datové typy.....	64
	Primitivní datové typy .....	64
	Objektové datové typy .....	65
	Vracení hodnot primitivních typů.....	65
	Vracení hodnot objektových typů .....	66
2.7	Parametry a jejich typy .....	68
	Vyvolání konstruktoru s parametry .....	69
	Parametry objektových typů.....	71
	Posílání zpráv s parametry .....	73
2.8	Metody třídy.....	73
2.9	Výlet do nitra instancí.....	75
	Atributy instancí.....	75
	Atributy třídy – statické atributy.....	77
2.10	Přímé zadávání hodnot parametrů objektových typů .....	79
	Veřejné atributy.....	79
	Odkazy vrácené po zaslání zprávy.....	81
2.11	Shrnutí – co jsme se naučili.....	83
<b>3.</b>	<b>Vytváříme vlastní třídu .....</b>	<b>85</b>
3.1	První vlastní třída.....	86
3.2	Zdrojový kód třídy.....	87
	Prázdná třída .....	88
	Bílé znaky a uspořádání programu .....	89
3.3	Soubory projektu .....	91
3.4	Odstranění třídy .....	92
3.5	Implicitní konstruktor .....	94
3.6	Přejmenování třídy.....	97
3.7	Ladění .....	98
	Syntaktické chyby.....	99
	Běžové chyby.....	100
	Logické (sémantické) chyby .....	103
3.8	Konstruktor s parametry .....	103
	Použití skrytého parametru this.....	105
3.9	Přetěžování .....	107
3.10	Testování .....	108
	TDD – vývoj řízený testy.....	108
	Zprovoznění nástrojů pro automatizaci testů.....	109

Testovací třída .....	110
Přípravek .....	111
Úprava obsahu přípravku .....	112
3.11 Shrnutí – co jsme se v kapitole naučili .....	114
Zdrojový kód .....	114
Ladění .....	114
Konstruktory a metody .....	115
<b>4. Přidáváme atributy a metody .....</b>	<b>116</b>
4.1 Deklarace atributů .....	116
Modifikátory přístupu .....	117
Vylepšujeme třídu Strom .....	118
Možné důsledky zveřejnění atributů .....	119
4.2 Definujeme vlastní metodu .....	120
Test vytvořených metod .....	121
Reakce na chybu v testu .....	124
Nejprve testy, pak program? .....	125
Někdy jsou věci složitější .....	127
Použití metod vracejících hodnotu .....	128
Definice metod vracejících hodnotu .....	129
Parametry a návratové hodnoty objektových typů .....	130
4.3 Doplnění projektu o třídu odjinud .....	130
4.4 Přístupové metody .....	131
Atributy versus vlastnosti .....	132
Konvence pro názvy přístupových metod .....	132
4.5 Kvalifikace a klíčové slovo this .....	133
Kvalifikace metod .....	133
Kvalifikace atributů .....	134
4.6 Atributy a metody třídy (statické atributy a metody) .....	136
Atributy třídy .....	136
Metody třídy .....	137
4.7 Čtení chybových hlášení .....	139
4.8 Lokální proměnné .....	140
4.9 Konstanty a literály .....	143
Konstanty objektových typů .....	145
Správná podoba literálů .....	145
boolean .....	146
int .....	146
double .....	146
String .....	147
null .....	147
4.10 Zapouzdření a skrývání implementace .....	148
Rozhraní versus implementace .....	148
Signatura × kontrakt .....	149
4.11 Komentáře a dokumentace .....	150
Proč psát srozumitelné programy .....	150
Tři druhy komentářů .....	151
Zakomentování a odkomentování části programu .....	152
Pomocné značky pro tvorbu dokumentace .....	152
Okomentování třídy Strom .....	153
Uspořádání jednotlivých prvků v těle třídy .....	160
Prázdná standardní třída .....	161
BlueJ a komentářová nápověda .....	163
Automaticky generovaná dokumentace .....	164
Dokumentace celého projektu .....	164
4.12 Třída Object .....	168
Anotace @Override .....	169

4.13	Metoda toString()	169
	Sčítání řetězců	169
	Jak definovat metodu toString()	170
4.14	Závěrečný příklad – UFO	171
	Předběžné poznámky	171
	Stručný přehled	171
	Třída Dispečer	172
	Jednodušší varianta	173
	Varianta ovládaná z klávesnice	173
	Třída UFO	174
	Atributy	174
	Konstruktor	174
	Metoda setRychlost(int, int)	174
	Metody getX(), getY(), getXRychlost(), getYRychlost(), getXTah(), getYTah()	174
	Metoda nakresli()	175
	Metoda popojed(int)	175
	Metody vpravo(), vlevo(), vzhůru(), dolů(), vypniMotory()	175
	Metoda toString()	176
	Třída UFO_4	176
	Třída UFOTest	176
4.15	Vytvoření samostatné aplikace	177
	Prohlížení obsahu JAR-souborů	177
	Třída spouštějící aplikaci	177
	Vytvoření souboru JAR s aplikací	178
	Stěhování projektu mezi platformami	179
	Problémy s kódováním znaků	180
4.16	Shrnutí – co jsme se v kapitole naučili	181
	Zdrojový kód	181
	Atributy a lokální proměnné	181
	Dokumentace	182
	Aplikace	183
5.	Dotváříme vlastní třídu	184
5.1	Jednoduché vstupy a výstupy	184
	Textové řetězce	184
	Rozdíl mezi prázdným řetězcem a null	186
	Čísla	186
5.2	Podrobnosti o operátorech	187
	Binární aritmetické operátory + - * / %	188
	Sčítání, odčítání, násobení	188
	Slučování řetězců +	189
	Dělení /	189
	Zbytek po dělení (dělení modulo) %	190
	Unární operátory + -	190
	Kulaté závorky ()	190
	Přiřazovací operátor =	191
	Složené přiřazovací operátory +=, -=, *=, /=, %=	191
	Operátor přetypování (typ)	192
	Univerzální přetypování na String	193
5.3	Počítáme instance	194
5.4	Inkrementační a dekrementační operátory	195
	Způsoby předávání hodnot	198
	Jiný způsob inicializace rodného čísla	199
5.5	Standardní výstupy	199
	Standardní chybový výstup	201
5.6	V útrokách testovací třídy	202
	Přípravek	204

Automaticky generované testy .....	206
Vlastní testy .....	206
Úklid .....	207
Metody assertEquals a assertTrue .....	207
Test testů .....	208
5.7 Debugger a práce s ním .....	210
Krokování programu .....	211
Okno debuggeru .....	214
Vlákna .....	214
Atributy třídy .....	214
Atributy instancí .....	215
Lokální proměnné .....	215
Pořadí volání – zásobník návratových adres .....	215
Krokování konstruktoru .....	216
Atributy a proměnné objektových typů .....	216
Už nezastavuj – ruším zarážky .....	217
Předčasný konec programu .....	217
Pozastavení běžícího programu .....	218
5.8 Hodnotové a odkazové objektové typy .....	218
Odkazové datové typy .....	219
Hodnotové typy .....	219
Program demonstrující rozdíl .....	220
Hodnotové typy podruhé .....	221
5.9 Projekt Zlomky .....	222
5.10 Metoda equals(Object) .....	223
Operátor porovnání == .....	224
Operátor logické konjunkce && .....	224
Operátor instanceof .....	224
Definice equals(Object) pro Zlomek .....	225
5.11 Shrnutí – co jsme se naučili .....	225
<b>6. Návrhové vzory .....</b>	<b>229</b>
6.1 Knihovni třída (Utility) .....	230
6.2 Přeppravka (Messenger) .....	230
6.3 Tovární metoda (Factory method) .....	233
6.4 Jedináček (Singleton) .....	234
6.5 Výčtové typy .....	237
6.6 Návrhový vzor Prázdný objekt (Null Object) .....	240
6.7 Shrnutí – co jsme se naučili .....	241
<b>Část 2: Více tváří .....</b>	<b>243</b>
<b>7. Rozhraní .....</b>	<b>244</b>
7.1 Návrhový vzor Prostředník (Mediator) .....	244
7.2 Kreslíme jinak .....	245
7.3 Syntaxe rozhraní .....	246
7.4 Instance rozhraní .....	247
7.5 Nový projekt .....	248
Práce s novým plátnem .....	251
7.6 Událostmi řízené programování .....	253
7.7 Implementace rozhraní .....	253
Implementace rozhraní v diagramu tříd .....	254
Odvolání implementace rozhraní .....	254
Implementace rozhraní ve zdrojovém kódu .....	255
7.8 Úprava zdrojového kódu třídy Strom .....	256
Třída musí jít přeložit .....	256

Testování .....	259
Opomenuté testy.....	263
Efektivita vykreslování .....	264
Závěrečné úpravy .....	264
Uložení odkazu na správce plátna do atributu třídy.....	264
Odstranění statického atributu krok .....	265
Úpravy posunových metod .....	265
Zefektivnění přesunu .....	265
Vnořený blok.....	266
<b>7.9 Implementace několika rozhraní .....</b>	<b>266</b>
<b>7.10 Návrhový vzor Služebník (Servant).....</b>	<b>267</b>
Proč zavádíme rozhraní.....	268
Implementace .....	269
Aplikace na náš projekt.....	269
Závěrečný test.....	270
<b>7.11 Refaktorování .....</b>	<b>272</b>
Ukázka.....	272
1. krok: Vytvoření testu .....	273
2. krok: Doplnění prázdných verzí testovaných metod .....	274
3. krok: Definice nových atributů.....	274
4. krok: Kopírování těla konstrukturu do těla metody .....	275
5. krok: Dočasné „odkonstantnění“ některých atributů .....	275
6. krok: Definice potřebných lokálních proměnných .....	275
7. krok: Odstranění tvorby nových instancí koruny a kmene .....	276
8. krok: Vrácení koruny a kmene mezi konstanty .....	276
9. krok: Vyvolání metody setRozměr(int,int) v konstrukturu .....	276
10. krok: Odstranění zdvojeného kódu z konstrukturu .....	277
11. krok: Doplnění metody setRozměr(Rozměr) .....	277
12. krok: Doplnění metody setOblast(Oblast).....	278
<b>7.12 Projekt Výtah .....</b>	<b>278</b>
<b>Analýza problému .....</b>	<b>279</b>
Okolí.....	279
Konstruktory .....	279
Potřebné metody.....	280
<b>Implementace .....</b>	<b>281</b>
Implementovaná rozhraní .....	281
Atributy.....	281
Postup při návrhu metod.....	282
Metoda doPatra(int) .....	282
Metoda přijedK(IPosuvný) .....	282
Metoda nástup(IPosuvný) .....	283
Metody výstupVpravo() a výstupVlevo() .....	283
Test převozu pasažéra .....	283
Metody odvezVpravo(IPosuvný,int) a odvezVlevo(IPosuvný,int) .....	284
<b>7.13 Shrnutí – co jsme se naučili.....</b>	<b>284</b>
<b>8. Budete si to přát zabalit? .....</b>	<b>286</b>
<b>8.1 Velké programy a jejich problémy.....</b>	<b>286</b>
<b>8.2 Balíčky .....</b>	<b>287</b>
Podbalíčky .....	288
Uspořádání podbalíčků s programy k dosavadní části knihy .....	288
Názvy tříd.....	289
<b>8.3 Balíčky a BlueJ.....</b>	<b>289</b>
Příprava stromu balíčků pro BlueJ ve správci souborů .....	289
Příprava stromu balíčků v BlueJ .....	290
Vytvoření struktury balíčků pro tuto kapitulu .....	290
Putování stromem balíčků .....	291
Odstraňování balíčků .....	291

	Zavírání a otevírání projektů .....	292
<b>8.4</b>	<b>Naplňujeme balíčky</b> .....	<b>292</b>
	Automatické vložení příkazu package .....	294
<b>8.5</b>	<b>Balíčky a příkaz import</b> .....	<b>295</b>
	Import celého balíčku .....	297
	Import a podbalíčky .....	297
	Balíček java.lang .....	297
	Změna balíčku .....	298
<b>8.6</b>	<b>Názvy balíčků</b> .....	<b>298</b>
<b>8.7</b>	<b>Příkazový panel</b> .....	<b>299</b>
	Nevýhody koncepce balíčků v BlueJ .....	299
	Zobrazení příkazového panelu .....	299
	Použití příkazového panelu .....	300
	Opakované používání příkazů .....	301
<b>8.8</b>	<b>Přístupová práva v rámci balíčku</b> .....	<b>301</b>
<b>8.9</b>	<b>Neveřejné třídy</b> .....	<b>302</b>
<b>8.10</b>	<b>Tvorba vlastních aplikací</b> .....	<b>303</b>
<b>8.11</b>	<b>Statický import</b> .....	<b>303</b>
<b>8.12</b>	<b>Shrnutí – co jsme se naučili</b> .....	<b>304</b>
<b>9.</b>	<b>Co takhle něco zdědit?</b> .....	<b>307</b>
<b>9.1</b>	<b>Co to je, když rozhraní dědí?</b> .....	<b>308</b>
<b>9.2</b>	<b>Jak to zařídit</b> .....	<b>308</b>
	Duplicitně deklarovaná implementace .....	309
<b>9.3</b>	<b>Společný potomek několika rozhraní</b> .....	<b>310</b>
	Třída Oblast a rozhraní IHýbací .....	312
<b>9.4</b>	<b>Návrhový vzor Stav (State)</b> .....	<b>313</b>
	Projekt Šipky .....	314
	Shrnutí .....	317
<b>9.5</b>	<b>Návrhový vzor Zástupce (Proxy)</b> .....	<b>318</b>
<b>9.6</b>	<b>Projekt Kabina</b> .....	<b>320</b>
	Předpřipravené třídy .....	320
	Třída rup.česky.tvary.Multipřesouvač .....	320
	Rozhraní rup.česky.tvary.IMultiposuvný .....	321
	Rozhraní doprava.IZastávka .....	321
	Třída doprava.Linka .....	321
	Úloha: třída doprava.Kabina .....	322
<b>9.7</b>	<b>Shrnutí – co jsme se naučili</b> .....	<b>323</b>
<b>10.</b>	<b>Dědit mohou i třídy</b> .....	<b>324</b>
<b>10.1</b>	<b>Podtřídy a nadtřídy</b> .....	<b>324</b>
	Specializace .....	324
	Zobecnění .....	325
	Realizace v OOP .....	325
	Univerzální (pra)rodič Object .....	326
<b>10.2</b>	<b>Experimenty s dědičností</b> .....	<b>327</b>
	Atributy a bezparametrické konstruktory tříd v projektu .....	328
	Hierarchie dědičnosti .....	329
	Podobjekt rodičovské třídy .....	331
	Explicitní volání konstruktoru předka .....	333
	Dosažitelnost parametru this .....	336
	Postup budování instance .....	336
	Chráněné atributy – modifikátor přístupu protected .....	337
	Dědičnost a metody tříd .....	337
	Metody instancí, jejich dědění a překryvání .....	338
	Nové metody .....	339
	Nepřekryté zděděné metody .....	339

Překryté zděděné metody .....	339
<b>Test chování překrývajících a překrytých metod .....</b>	<b>340</b>
Porovnání .....	342
Podobnost .....	343
Soukromá metoda .....	343
Veřejná metoda .....	343
Instance vnučka .....	343
Vyvolání překryté verze metody .....	344
<b>10.3 Vytváříme dceřinou třídu .....</b>	<b>345</b>
Jednoduchá dceřiná třída .....	346
Konstruktory potomka .....	347
Složitější dceřiná třída .....	348
Definice konstruktorů .....	348
Metoda kresli(Kreslítko) .....	349
Metoda setPozice(int,int) .....	350
Jak přesvědčit objekt, aby se pokaždé choval jinak .....	352
Samostatná úloha: Terč .....	353
<b>10.4 Vytváříme rodičovskou třídu .....</b>	<b>356</b>
Společný rodič Posuvný .....	356
Příprava .....	356
Konstantní atributy třídy .....	357
Proměnné atributy třídy .....	357
Konstantní atributy instancí .....	358
Proměnné atributy instancí .....	358
Konstruktory .....	359
Metody instancí .....	360
Třídy jako objekty – class-objekt třídy .....	361
Doladění dceřiných tříd .....	362
Elipsa, Obdélník, Trojúhelník .....	362
Čára .....	362
Text .....	363
Strom .....	364
Dodatečné rozšíření rodičovské třídy .....	364
Společný rodič Hýbací .....	366
<b>10.5 Abstraktní metody a třídy .....</b>	<b>367</b>
Neimplementovaná metoda implementovaného rozhraní .....	368
Zděděná a neimplementovaná abstraktní metoda .....	368
Přidání metody zobraz() .....	369
Nově deklarovaná abstraktní metoda .....	369
Abstraktní třída bez abstraktních metod .....	370
<b>10.6 Nová schopnost – přizpůsobivost .....</b>	<b>370</b>
<b>10.7 Návrhový vzor Stav podruhé .....</b>	<b>371</b>
Projekt Šípka .....	372
<b>10.8 Co je na dědičnosti špatné .....</b>	<b>374</b>
<b>10.9 Třída ZpětnáKabína .....</b>	<b>374</b>
<b>10.10 Omezení kladená na konstruktory .....</b>	<b>376</b>
<b>10.11 Konečné třídy .....</b>	<b>377</b>
Poznámka o dobrých mravech .....	378
<b>10.12 Konečné metody .....</b>	<b>379</b>
<b>10.13 ZpětnáKabína podruhé .....</b>	<b>380</b>
<b>10.14 Tovární metoda podruhé .....</b>	<b>381</b>
Jak něco udělat před spuštěním rodičovského konstrukturu .....	381
Využití tovární metody .....	382
<b>10.15 Kdy (ne)použít dědičnost .....</b>	<b>383</b>
Potomci, kteří nejsou speciálním případem rodiče .....	383
Kdy jsme použili dědičnost místo správného skládání .....	384
Potomci, kteří jsou příliš speciální .....	385



	Kdy dát přednost skládání a kdy dědičnosti .....	386
	10.16 Shrnutí – co jsme se naučili .....	386
<b>11.</b>	<b>Knihovny .....</b>	<b>389</b>
	11.1 Zbývající primitivní datové typy .....	389
	long .....	389
	short .....	390
	byte .....	390
	float .....	391
	char .....	391
	11.2 Primitivní a obalové datové typy .....	392
	11.3 Třída System .....	393
	11.4 Formátovaný výstup .....	393
	Národní prostředí .....	394
	Ukázka .....	395
	11.5 Základní matematické funkce .....	395
	11.6 Pracujeme s náhodou .....	396
	11.7 Ukončení aplikace .....	398
	11.8 Třída String .....	399
	11.9 Definice vlastní knihovny a její začlenění do BlueJ .....	399
	Vytvoření JAR-souboru s knihovnou .....	400
	Přidání knihovny do BlueJ .....	401
	11.10 Shrnutí – co jsme se naučili .....	402
<b>12.</b>	<b>Program ve výjimečné situaci .....</b>	<b>403</b>
	12.1 Nejdůležitější výjimky .....	404
	12.2 Vyhození výjimky .....	405
	Výjimky a dostupný kód .....	406
	12.3 Co výjimky umí .....	406
	getMessage() .....	406
	toString() .....	407
	printStackTrace() .....	407
	printStackTrace( PrintStream ) .....	407
	12.4 Zachycení vyhozené výjimky .....	407
	Analýza rekurzivní metody .....	408
	Několik současně odchyťovaných výjimek .....	409
	Společný úklid .....	410
	Testování správného vyhození výjimky .....	411
	12.5 Hierarchie dědičnosti výjimek .....	412
	Definice vlastních výjimek .....	413
	Kontrolované výjimky .....	414
	Převod kontrolovaných výjimek na nekontrolovanou .....	415
	12.6 Shrnutí – co jsme se naučili .....	417

## **Část 3: Učíme program přemýšlet 419**

<b>13.</b>	<b>Program začíná přemýšlet .....</b>	<b>420</b>
	13.1 Ternární operátor ?: .....	420
	13.2 Jednoduchý podmíněný příkaz .....	421
	Vyhození výjimky .....	424
	13.3 Blok příkazů (složený příkaz) .....	425
	13.4 Podmínky a jejich skládání .....	426
	Porovnávací operátory .....	426
	Logické výrazy .....	427
	Použití v programu .....	428
	13.5 Návrhový vzor Adaptér (Adapter) .....	429