

edice

začínáme s ...

PROGRAMOVÁNÍ V JAZYKU JAVA

Učebnice pro budoucí profesionální programátory

- » Přehled základních jazykových konstrukcí a principů OOP
- » Návrh objektové architektury pro škálovatelné aplikace
- » Tvorba základního rámce projektu a jeho API pro efektivní práci týmu
- » Zefektivnění vývoje pomocí testovacích scénářů

📄 soubory ke stažení na www.grada.cz



edice
začínáme s ...

Programování v jazyku JAVA

**Učebnice pro budoucí
profesionální programátory**

RUDOLF PECINOVSKÝ

Rudolf Pecinovský

Programování v jazyku Java

Učebnice pro budoucí profesionální programátory

Vydala Grada Publishing, a.s.

U Průhonu 22, Praha 7

obchod@grada.cz, www.grada.cz

tel.: +420 234 264 401

jako svou 10132. publikaci

Odpovědný redaktor Petr Somogyi

Fotografie na obálce Depositphotos/KostyaKlimenko

Grafická úprava a sazba Rudolf Pecinovský

Počet stran 576

První vydání, Praha 2025

Vytiskly Tiskárny Havlíčkův Brod, a. s.

© Grada Publishing, a.s., 2025

Cover Design © Grada Publishing, a. s., 2025

Cover Photo © Depositphotos/KostyaKlimenko

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.

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

Automatizovaná analýza textů nebo dat ve smyslu čl. 4 směrnice 2019/790/EU a použití této knihy k trénování AI jsou bez souhlasu nositele práv zakázány.

ISBN 978-80-271-8054-7 (ePub)

ISBN 978-80-271-8053-0 (pdf)

ISBN 978-80-271-5751-8 (print)

Věnováno mé ženě Jarušce

Stručný obsah

Poděkování	22
Úvod	23
O autorovi	30
Část A Superzáklady	31
Kapitola 1 Předehra	32
Kapitola 2 Prostředí JShell	45
Kapitola 3 Zadávaní hodnot	60
Kapitola 4 Proměnné, výrazy a příkazy	74
Část B Začínáme programovat	91
Kapitola 5 Práce s objekty	92
Kapitola 6 Balíčky, knihovny, robot Karel	103
Kapitola 7 Definice metod	119
Kapitola 8 Opakování kódu – cykly	140
Kapitola 9 Rozhodování	152
Část C Objektově orientované programování	171
Kapitola 10 Základy definice třídy	172
Kapitola 11 Uspořádání kódu	188
Kapitola 12 Programátorská dokumentace	202
Kapitola 13 Rozhraní a interfejsy	211
Kapitola 14 Dědění interfejsů	222
Kapitola 15 Návrhové vzory	236
Kapitola 16 Prohlubujeme znalosti	252
Kapitola 17 Lambda-výrazy, funkční interfejsy a generické typy a metody	274
Kapitola 18 Dědění implementace	292
Kapitola 19 Abstraktní třídy	312

Část D Standardní knihovna 329

Kapitola 20 Speciální datové typy.....	330
Kapitola 21 Výjimky.....	348
Kapitola 22 Kontejnery	362
Kapitola 23 Pole	377
Kapitola 24 Interní datové typy	391
Kapitola 25 Datovody	405
Kapitola 26 Čtení a ukládání dat	416
Kapitola 27 Vytváření aplikací	434

Část E Vývoj aplikace 445

Kapitola 28 Zásady objektové architektury.....	446
Kapitola 29 Návrh základní architektury	455
Kapitola 30 Testovací scénáře – dokončení API	471
Kapitola 31 První a druhá etapa: scénáře HAPPY a BASIC	482
Kapitola 32 Třetí etapa: Implementace navržené architektury	494
Kapitola 33 Čtvrtá etapa: spuštění hry	505
Kapitola 34 Pátá etapa: tvorba světa	514
Kapitola 35 Šestá etapa: definice standardních akcí	523
Kapitola 36 Sedmá a osmá etapa: děláme aplikaci robustní	530
Kapitola 37 Etapy 9–11: přidáváme nestandardní akce	542
Kapitola 38 Závěr: doplnění uživatelského rozhraní	555
Literatura.....	568
Rejstřík.....	571

Část F Přílohy 577

Příloha A Příprava spuštění JShell pod Windows.....	578
Příloha B Význam cizích slov	583
Příloha C Česko-americký slovník.....	585

Část G Seznamy 588

Seznam S1 Výpisy programů	589
Seznam S2 Seznam obrázků	595
Seznam S3 Seznam tabulek	597
Seznam S4 Seznam odboček – podšeděných bloků.....	598

Podrobný obsah

Poděkování	22
Úvod	23
Komu je kniha určena	23
Koncepce výkladu	24
Potřebné vybavení	25
Doprovodné programy	26
Použité typografické konvence	27
Odbočka – podšeděný blok	29
Zpětná vazba	29
O autorovi	30
 Část A Superzáklady	 31
Kapitola 1 Předehra	32
1.1 Trocha historie	32
1.1.1 Co je to program	33
1.1.2 Postupná změna priorit	33
1.1.3 Vývoj metodik programování	35
1.1.4 OOP a OFP	36
1.2 Překladače, interprety, platformy	36
1.2.1 Operační systém a platforma	37
1.2.2 Programovací jazyky	37
1.2.3 Způsoby zpracování programu	38
1.3 Java a její zvláštnosti	39
1.3.1 Java je jazyk i platforma	40
1.4 Vývojové nástroje	41
1.4.1 Základní vývojářská sada	41
1.4.2 IDE	41
1.5 Shrnutí a soubory pro opakování	44
Kapitola 2 Prostředí JShell	45
2.1 Prostředí JShell	45
2.1.1 Spuštění programu JShell	45
2.1.2 JShell v IDE	47
2.2 Úryvky (snippets)	48
2.2.1 Použití proměnných	49
2.2.2 Identifikace úryvků	49
2.2.3 Terminologie: výrazy, příkazy, deklarace, definice	50
2.2.4 Středník	50
2.2.5 Více objektů na řádku, zavlečené chyby	50
2.3 Příkazy prostředí JShell	52
2.3.1 Vyloučení úryvku: /drop	52
2.3.2 Přehled aktivních úryvků: /list	52

2.3.3 Přehled všech úryvků: <code>/list -all</code>	53
2.3.4 Uložení aktivních úryvků: <code>/save <file></code>	54
2.3.5 Uložení všech zadaných úryvků: <code>/save -all <file></code>	55
2.3.6 Uložení dosavadního průběhu seance: <code>/save -history <file></code>	55
2.3.7 Načtení skriptu: <code>/open <file></code>	55
2.3.8 Ukončení seance: <code>/exit</code>	56
2.3.9 Restart: <code>/reset</code>	56
2.3.10 Znovuzavedení: <code>/reload -restore</code>	57
2.3.11 Natavení startovního skriptu: <code>/set -start <file></code>	57
2.3.12 Náповěda: <code>/?</code>	58
2.3.13 Spuštění editoru: <code>/edit</code>	58
2.3.14 Nastavení běhového prostředí: <code>/env</code>	59
2.4 Shrnutí a soubory pro opakování	59
Kapitola 3 Zadávání hodnot	60
3.1 Datové typy	60
3.1.1 Primitivní datové typy	61
3.1.2 Objektové datové typy	62
3.2 Komentáře	63
3.3 Zadávání celočíselných hodnot	64
3.4 Zadávání reálných hodnot	64
3.5 Zadávání znaků	66
3.6 Prázdný odkaz <code>null</code>	68
3.7 Hodnoty typu <code>String</code>	68
3.7.1 Textové bloky	71
3.8 Zadávání logických hodnot	72
3.9 Literály	73
3.10 Shrnutí a soubory pro opakování	73
Kapitola 4 Proměnné, výrazy a příkazy	74
4.1 Pravidla pro tvorbu identifikátorů	74
4.1.1 Používání znaku <code>\$</code>	75
4.2 Deklarace proměnných	76
4.3 Proměnné versus konstanty	77
4.4 Výrazy a jejich terminologie	78
4.5 Operace přiřazení =	79
4.5.1 Asociativita	80
4.6 Aritmetické operace	80
4.6.1 Sčítání	80
4.6.2 Odčítání	80
4.6.3 Násobení	81
4.6.4 Dělení	81
4.7 Přetypování	81
4.8 Volání metod	84
4.9 Závorky	84
4.10 Složené přiřazení	85
4.11 Inkrementační a dekrementační operátory	86
4.12 Porovnávací operátory	88
4.13 Příkazy versus výrazy	89
4.14 Shrnutí a soubory pro opakování	89

Část B Začínáme programovat **91**

Kapitola 5 Práce s objekty	92
---	-----------

5.1 Nejprve trocha teorie.....	92
5.1.1 Principy OOP	92
5.1.2 Objekty	93
5.1.3 Atributy	93
5.1.4 Zprávy	94
5.1.5 Metody	94
5.1.6 Třídy a jejich instance	95
5.1.7 Interní datové typy	96
5.1.8 Třída jako datový typ	96
5.1.9 Třída jako objekt	96
5.1.10 Kontejner	97
5.1.11 Terminologie objektových datových typů	97
5.2 Práce s atributy	98
5.3 Volání metody	98
5.4 Konstruktory a tovární metody	100
5.4.1 Tovární metoda	100
5.4.2 Konstruktory	101
5.4.3 Správa paměti	101
5.5 Shrnutí a soubory pro opakování	102
Kapitola 6 Balíčky, knihovny, robot Karel	103
6.1 Velké programy a jejich problémy	103
6.2 Balíčky	104
6.2.1 Kořenový balíček a stromy balíčků	104
Datová struktura strom	105
6.3 Import objektových typů	105
6.3.1 Import všech typů z daného balíčku – hvězdičkový import	107
6.4 Knihovny a JAR-soubory	108
6.4.1 Začlenění knihovny robota Karla	108
6.4.2 Začlenění knihovny při startu	110
6.5 Robot Karel a jeho svět	110
Historie robota Karla	111
6.5.1 Vytvoření světa	111
6.5.2 Vytvoření robota	113
6.5.3 Akce	113
6.5.4 Testy	114
6.5.5 Zrychlování a skrývání	114
6.5.6 Reakce na zavření okna	116
6.6 Zakázaný balíček java	118
6.7 Shrnutí a soubory pro opakování	118
Kapitola 7 Definice metod	119
7.1 Start JShell na počátcích dalších kapitol	119
7.2 Zásada DRY	120
7.3 Definice a volání metody	120
7.3.1 Povinný středník	121
7.3.2 Příklad jednoduché metody pro robota	121
7.4 Blok příkazů	122
7.5 Metody s parametry	123
7.5.1 Parametry	123
7.5.2 Argumenty	123
7.5.3 Metody robota versus metody interaktivního prostředí	124
7.6 Metody s více parametry	125
Tisk na standardní výstup	126
7.7 Lokální proměnné a konstanty	126
7.7.1 Proměnné lokální v bloku	127
7.7.2 Demonstrace použití konstant	127

7.7.3 Dočasná existence proměnných v blocích	128
Zásobník návratových adres – ZNA	128
7.8 Přetěžování metod	129
7.9 Ještě jednou robot Karel a jeho svět	130
7.9.1 Třídy RobotWorld a RobotWindow	130
7.9.2 Přetížené tovární metody tříd RobotWorld a RobotWindow	131
7.9.3 Přetížené konstruktory třídy Karel	132
7.10 Metody vracející hodnotu	133
7.11 Přehled aktuálně definovaných metod v JShell	134
7.12 Logické výrazy	135
7.12.1 Příklad	136
7.13 Předávání hodnotou a odkazem	137
7.14 Shrnutí a soubory pro opakování	139
Kapitola 8 Opakování kódu – cykly	140
8.1 Cykly	140
8.2 Cyklus s počáteční podmínkou – cyklus while	141
8.3 Cyklus s koncovou podmínkou – cyklus do...while	143
8.4 Cyklus s parametrem – cyklus for	145
8.4.1 Příklad	146
8.5 Nekonečný cyklus	149
8.6 Cyklus s podmínkou uprostřed	149
8.7 Vnořování cyklů	150
8.8 Shrnutí a soubory pro opakování	151
Kapitola 9 Rozhodování	152
9.1 Jednoduchý podmíněný příkaz	153
9.2 Úplný podmíněný příkaz	154
9.3 Podmíněný výraz	155
9.4 Složený podmíněný příkaz	156
9.5 Přepínač – příkaz/výraz switch	158
9.5.1 Pravidla	160
9.5.2 Přepínací výraz – výraz switch	162
9.5.3 Využití klasické verze příkazu switch	163
9.6 Cyklus s podmínkou uprostřed – příkaz break	164
9.7 Příkaz continue	166
9.8 Rekurze	167
9.9 Shrnutí a soubory pro opakování	169

Část C Objektově orientované programování **171**

Kapitola 10 Základy definice třídy	172
10.1 Vytváříme vlastní třídu	172
Bílé znaky a uspořádání programu	173
10.2 Viditelnost tříd a jejich členů: public , private	174
10.3 Výměna knihoven – knihovna tvarů	174
10.4 Diagramy tříd	176
10.5 Konstruktory	177
10.5.1 Podrobnosti o konstruktorech	177
10.6 Atributy	179
10.6.1 Definice atributů	179
10.6.2 Modifikátor final	179
10.7 Konstruktory a parametr this	180
10.7.1 Kvalifikace parametrem this	181

10.8	Magické hodnoty	182
10.9	Modifikátor static	183
10.9.1	Definice třídy počítající své instance	183
10.10	Zděděné metody, metoda toString()	184
10.11	Výsledná definice a test.....	184
10.11.1	Upravená definice	185
10.11.2	Test upravené definice	185
10.12	Shrnutí a soubory pro opakování	187
Kapitola 11	Uspořádání kódu	188
11.1	Uspořádání programů ve složce 77_JAVA	188
11.1.1	Vývojové prostředí	189
11.1.2	Problémy s překladem	189
11.1.3	Knihovna Java77_NZ2	189
11.1.4	Úprava prostředí pro JShell	190
11.2	Konvence pro názvy balíčků v Javě	190
11.3	Samostatně definované třídy	190
11.4	Příkazy package a import	192
11.5	Zapouzdření a skrývání implementace	193
11.6	Entity programu	194
11.7	Rozhraní versus implementace.....	194
11.7.1	Signatura versus kontrakt	195
11.7.2	API.....	195
11.8	Atributy versus vlastnosti; přístupové metody	196
11.8.1	Možné kombinace a výhody	196
11.8.2	Pravidla pro názvy.....	197
11.9	Uspořádání jednotlivých prvků v těle třídy	197
11.9.1	Motivace pro zavedení některých zásad	198
11.10	Prázdná standardní třída.....	199
11.11	Shrnutí a soubory pro opakování	201
Kapitola 12	Programátorská dokumentace.....	202
12.1	Komentáře a dokumentace.....	202
12.1.1	Proč psát srozumitelné programy	202
12.1.2	Odsazování	204
12.1.3	Dokumentační komentáře	204
12.2	Pomocné značky pro tvorbu dokumentace	205
12.3	Dokumentace balíčku	206
12.4	Ukázka dokumentované třídy	206
12.5	Zobrazení dokumentace.....	208
12.5.1	Obsah a uspořádání dokumentace.....	208
12.6	Zakomentování a odkomentování části programu	210
12.7	Shrnutí a soubory pro opakování	210
Kapitola 13	Rozhraní a interfejs	211
13.1	Motivace	211
13.2	Rozhraní versus interfejs versus interface	212
13.2.1	Interfejs a jeho instance	212
13.3	Použití v programu	213
13.3.1	Knihovní balíček shapes77.canvas_2	214
13.4	Použití interfejsu na příkladu.....	215
13.4.1	Počáteční úvahy	215
13.5	Definice interfejsu.....	217
13.6	Implementace interfejsu třídou	218
13.6.1	Anotace @Override	218
13.6.2	Ověření funkcionality	220

13.7 Shrnutí a soubory pro opakování	221
Kapitola 14 Dědění interfejsů	222
14.1 Problém více rolí	222
14.2 Současná implementace více interfejsů	223
14.2.1 Realizace v kódu	224
14.3 Dědění interfejsů	225
14.3.1 Trocha teorie o dědění	225
14.3.2 Dědění a přetypování	226
14.3.3 Aplikace dědění interfejsů – balíček canvas_4	227
14.4 Deklarace rodičů interfejsu v kódu	229
14.5 Vlastnosti interfejsů na příkladu Multishape	229
14.5.1 Podrobnosti o třídě Multishape	230
14.5.2 Mnohotvar se skládá z kopií	230
14.6 Příklad: definice vozidla	230
14.6.1 Kontrakt	231
14.6.2 Test vytvořeného vozidla	234
14.6.3 Problém plátna	235
14.7 Shrnutí a soubory pro opakování	235
Kapitola 15 Návrhové vzory	236
15.1 Úvod do návrhových vzorů	236
15.2 Přehled vzorů, s nimiž jsme se již setkali	238
15.2.1 Knihovni třída (Utility class)	238
15.2.2 Statická tovární metoda (Static factory method)	238
15.2.3 Jedináček (Singleton)	239
15.2.4 Náhradník (Substitute)	239
15.2.5 Výčtový typ (Enumerated type)	239
15.2.6 Multiton, Originál	240
15.2.7 Služebník (Servant)	240
15.2.8 Prázdný objekt (Null Object)	241
15.2.9 Prototyp (Prototype)	241
15.3 Teoretické pozadí nové koncepce kreslení	242
15.3.1 Návrhový vzor Prostředník (Mediator)	242
15.3.2 Inverze závislostí	243
15.3.3 Návrhový vzor Pozorovatel (Observer), hollywoodský princip	245
15.4 Správce plátna – CanvasManager	246
15.4.1 Balíček canvasmanager	247
15.4.2 Import klíčových tříd knihovny	247
15.5 Nová verze třídy Robot	249
15.6 Shrnutí a soubory pro opakování	251
Kapitola 16 Prohlubujeme znalosti	252
16.1 Statický import	252
16.2 Další členy interfejsů	253
16.2.1 Statické konstanty	253
16.2.2 Statické metody	255
16.2.3 Implicitní metody	255
16.2.4 Soukromé metody	255
16.3 Návrhový vzor Adaptér (Adapter)	256
16.4 Návrhový vzor Převrátka – třídy typu record	257
16.4.1 Třídy typu záznam (record)	257
16.4.2 Záznamy v používané knihovně	258
16.5 Návrhový vzor Šablonová metoda	260
16.5.1 Příklad	261
16.6 Návrhový vzor Abstraktní továrna	262
16.6.1 Motivace	262

16.6.2	Návrhový vzor Tovární metoda.....	263
16.6.3	Co je abstraktní továrna	263
16.6.4	Použití.....	263
16.6.5	Tovární třída převádí konstruktory a statické členy na instanční.....	263
16.7	Podrobnosti o konstruktorech	265
16.7.1	Dvě části těla konstrukturu instancí – inicializační bloky.....	266
16.7.2	Inicializace konstant	268
16.7.3	Konstruktor třídy	268
16.7.4	Konstanty vyhodnotitelné v době překladu	271
16.8	Shrnutí a soubory pro opakování	273
Kapitola 17	Lambda-výrazy, funkční interfejsy a generické typy a metody	274
17.1	Nezávislé opakování zadané akce	274
17.1.1	Možná řešení.....	275
17.2	Syntaxe lambda-výrazů.....	275
17.2.1	Aplikace na světlo	276
17.3	Lambda-výrazy a lokální proměnné.....	278
17.4	Lambda-výrazy zastupující metody	279
17.5	Funkční interfejsy	280
17.6	Generické datové typy a metody.....	281
17.6.1	Motivace	281
17.6.2	Syntaxe zadávání a používání generických typů	281
17.6.3	Specifika generických typů Javy	282
17.6.4	Omezení typových parametrů	282
17.6.5	Příklad: Interval	282
17.6.6	Typové parametry s více předky.....	284
17.6.7	Potomci a předci generických typů	284
17.6.8	Generické metody	285
17.7	Funkční interfejsy – pokračování.....	285
17.7.1	Demonstrace použití	287
17.8	Lambda-výrazy nelze přetypovat na Object přímo	289
17.9	Shrnutí a soubory pro opakování	291
Kapitola 18	Dědění implementace	292
18.1	Tři druhy dědění.....	292
18.1.1	Přirozené (nativní) dědění	292
18.1.2	Dědění rozhraní	293
18.1.3	Dědění implementace	293
18.1.4	LSP – Liskov Substitution Principle.....	294
18.1.5	Shrnutí	294
18.2	Základy dědění tříd	294
18.2.1	Univerzální (pra)rodič Object.....	295
18.3	Co dědíme od třídy Object.....	295
18.3.1	Class-objekt	296
18.3.2	Úplný název tříd v JShell	297
18.3.3	Přehled veřejných členů zděděných od třídy Object	297
18.4	Definice třídy s předkem – Square	298
18.4.1	Rodičovský podobjekt.....	298
18.4.2	Volání rodičovského konstrukturu.....	299
18.5	Přebíjení metod.....	301
18.5.1	Implementace přebíjení	303
18.6	Skrytá nebezpečí: úprava třídy Square	305
18.7	Modifikátor final v procesu dědění	307
18.7.1	Virtuální versus konečné metody	307
18.7.2	Obyčejné versus konečné třídy	308
18.8	Zakrývání statických metod.....	308
18.8.1	Metody interfejsů se nezakrývají	309

18.9	Jediný implementační předek	309
18.10	Přístupová práva	310
18.11	Akceptovatelné modifikace signatur potomků	310
18.12	Shrnutí a soubory pro opakování	311
Kapitola 19	Abstraktní třídy	312
19.1	Abstraktní třídy a jejich role v dědické hierarchii	312
19.1.1	Definice abstraktních tříd a metod	314
19.1.2	Experimenty s abstraktní třídou	314
19.2	Účel abstraktních tříd	316
19.3	Zavedení abstraktních tříd do projektu	316
19.4	Návrhový vzor Stav	317
19.5	Aplikace na příklad s vozidlem	318
19.5.1	Hlavní třída vozidla – třída Robot4_4	319
19.5.2	Stavově nezávislé metody	322
19.5.3	Stavově závislé metody	322
19.5.4	Společný rodič jednosměrných tříd – třída ARobot1_4	322
19.5.5	Jednostavové (jednosměrné) třídy	324
19.5.6	Test	326
19.6	Shrnutí a soubory pro opakování	327

Část D Standardní knihovna **329**

Kapitola 20	Speciální datové typy	330
20.1	Třída Object	330
20.2	Objekty reprezentující hodnotu (ORV) a objekty reprezentující entitu (ORE)	331
20.2.1	Metody equals(Object)	332
20.2.2	Metoda hashCode()	333
20.2.3	Proměnné a neměnné ORV	333
20.3	Primitivní a obalové typy	335
20.4	Třída String	336
20.4.1	Možné problémy při práci s některými znaky	336
20.4.2	Interfejs java.lang.CharSequence	336
	Problémy s kódováním znaků	337
20.4.3	Třídy StringBuilder a StringBuffer	338
20.5	Výčtové typy – třídy typu enum	339
20.5.1	Složitější příklad: Direction	341
20.6	Třída Throwable	342
20.7	Třída Thread	342
20.8	Třída java.util.Optional<T> & spol.	343
20.8.1	Motivace	343
20.8.2	Alternativní řešení	343
20.9	Třída java.util.Random	344
20.10	Interfejs java.lang.Comparable<T>	345
20.11	Interfejs java.util.Comparator<T>	346
20.12	Shrnutí a soubory pro opakování	347
Kapitola 21	Výjimky	348
21.1	Co to jsou výjimky	348
21.2	Nejdůležitější výjimky	349
21.3	Vyhození výjimky	350
21.3.1	Reakce systému na vyhození výjimky	351
21.4	Výjimky a nedosažitelný kód	353
21.5	Hierarchie dědění výjimek	353

21.6	Zachycení vyhozené výjimky.....	355
21.7	Společný úklid – blok finally	356
21.8	Definice vlastních výjimek	358
21.9	Kontrolované výjimky	359
21.10	Převedení kontrolované výjimky na nekontrolovanou	360
21.11	Shrnutí a soubory pro opakování	361
Kapitola 22	Kontejnery	362
22.1	Co je to kontejner v Javě	362
22.2	Kategorizace kontejnerů	363
22.2.2	Kontejnery objektů	363
22.3	Knihovna kolekcí	364
22.4	Deklarujte typy co nejobecněji	366
22.5	Collection<E> – kolekce	367
22.6	Dvojtečkový cyklus for(:)	368
22.7	Množiny – Set<E>	369
22.8	Seznamy – List<E>	371
22.8.1	Pořadí prvků	371
22.8.2	Příklad: seřazení prvků v seznamu	373
22.9	Slovníky a mapy – Map<K, V>	374
22.9.1	Pohledy	376
22.10	Shrnutí a soubory pro opakování	376
Kapitola 23	Pole	377
23.1	Představení	377
23.1.1	Deklarace polí	378
23.2	Atributy a metody polí	378
23.3	Pole jako kontejner	378
23.3.1	Pole odkazů na objekty	379
23.3.2	Pole hodnot primitivních typů	379
23.3.3	Hlídání mezi polí	381
23.3.4	Inicializace polí v deklaraci	382
23.3.5	Inicializace vytvářeného pole	383
23.3.6	Neinicializovaná pole objektových typů	384
23.4	Vícerozměrná pole	385
23.4.1	Obdélníková pole	385
23.4.2	Neobdélníková pole	386
23.4.3	Inicializace vícerozměrného pole	387
23.5	Metody s proměnným počtem argumentů	387
23.6	Pole, kolekce a moderní programování	388
23.7	Závěrečný příklad	388
23.8	Shrnutí a soubory pro opakování	390
Kapitola 24	Interní datové typy	391
24.1	Přehled	391
24.1.1	Terminologie	391
24.1.2	Společné charakteristiky	392
24.1.3	Použití	392
24.1.4	Jedna hladina soukromí	393
24.2	Globální interní (členské) datové typy	393
24.3	Vnořené datové typy	394
24.4	Vnitřní třídy	395
24.5	Lokální třídy	395
24.5.1	Pojmenované lokální třídy	396
24.5.2	Anonymní třídy	396
24.6	Návrhový vzor Iterátor	397

24.6.1	Interfejsy <code>java.util.Iterator<E></code> a <code>java.lang.Iterable<E></code>	398
24.6.2	Vnější (sekvencní) a vnitřní (dávkové) iterátory	399
24.6.3	Iterátory a dvojtečkový cyklus <code>for(:)</code>	400
24.7	Definice vlastní iterovatelné třídy	401
24.7.1	Generátor náhodných stringů	402
24.7.2	Použití v cyklu <code>for(:)</code>	403
24.7.3	Použití interního iterátoru <code>forEach()</code>	404
24.8	Shrnutí a soubory pro opakování	404
Kapitola 25	Datovody	405
25.1	Analogie	405
25.2	Druhy operací	406
25.3	Princip práce datovodu	406
25.4	Specifické vlastnosti	407
25.5	Datové typy související s datovody	408
25.6	Vytváření datovodů	408
25.6.1	Kolekce	408
25.6.2	Pole	409
25.6.3	Interfejs <code>java.util.stream.Stream</code>	409
25.7	Konceptní příklad	409
25.8	Operace s daty v datovodu	410
25.8.1	Koncové operace	410
25.8.2	Průběžné operace	411
25.8.3	Částečně průběžné operace	412
25.9	Kolektory	412
25.9.1	Příklad	413
25.10	Shrnutí a soubory pro opakování	415
Kapitola 26	Čtení a ukládání dat	416
26.1	Koncepce čtení a ukládání dat	416
26.2	Soubory: bleskové opakování	417
26.2.1	Soubor, souborový systém, cesta	417
26.2.2	Relativní, absolutní a kanonická cesta	418
26.2.3	Substituované disky ve Windows	418
26.3	Třída <code>java.io.File</code>	419
26.3.1	Instanční metody	419
26.4	Návrhový vzor Dekorátor	422
26.4.1	Motivace	422
26.4.2	Princip funkce	423
26.5	Rozdělení datových proudů	424
26.6	Zápis textů	425
26.6.1	Splachování a zavírání proudů	426
26.6.2	Přidávání dat na konec existujícího souboru	427
26.6.3	Příklad	427
26.7	Čtení textů	429
26.7.1	Příklad	430
26.8	Třída <code>java.util.Scanner</code>	431
26.9	Shrnutí a soubory pro opakování	433
Kapitola 27	Vytváření aplikací	434
27.1	Superjednoduchá demonstrační aplikace	434
27.2	Základní pravidla pro větší aplikace	437
27.3	Hlavní třída aplikace	437
27.3.1	Argumenty příkazového řádku	438
27.3.2	Definice třídy <code>Main</code>	438
27.3.3	Definice třídy <code>GuessIO</code>	440

27.4 JAR-soubor	441
Prohlížení obsahu JAR-souborů	441
27.4.1 Soubor MANIFEST.MF	442
27.4.2 Vytvoření JAR-souboru s aplikací	442
27.5 Spuštění aplikace	443
27.6 Shrnutí a soubory pro opakování	444

Část E Vývoj aplikace 445

Kapitola 28 Zásady objektové architektury	446
28.1 Předmluva	446
28.2 Architektura	447
28.3 Hlavní zásady návrhu	447
28.3.1 Připravenost na změny	448
28.3.2 CRIDP – maximální přehlednost	448
28.3.3 KISS – maximální jednoduchost	448
28.3.4 YAGNI – žádné zbytečnosti	448
28.3.5 SoC – jediný zodpovědný	449
28.3.6 SRP – jediná zodpovědnost	449
28.4 Antivzory	449
28.5 Návrh programu	450
28.6 Druhy vytvářených sólo-objektů	451
28.7 Dva způsoby návrhu	452
28.7.1 Návrh shora dolů	452
28.7.2 Návrh zdola nahoru	452
28.7.3 Porovnání	453
28.8 UML – diagram tříd	454
28.9 Důležitost čitelnosti programu	454
28.10 Shrnutí a soubory pro opakování	454
Kapitola 29 Návrh základní architektury	455
29.1 Proč právě textová konverzační hra	455
29.1.1 Proč hra vyvíjená pod frameworkem	456
29.2 Organizace balíčků a záznamy průběhu vývoje	456
29.2.1 Umístění studentských úloh	457
29.2.2 Umístění tříd frameworku	457
29.3. Koncepce vyvíjené aplikace	457
Co to je h-objekt	458
29.3.1 Nestandardní akce	459
29.4 Zadání	459
29.4.1 Vyjmenované požadavky	460
29.5 Účastníci – objekty vystupující ve hře	461
29.6 Správci skupin objektů	465
29.6.1 Správci v naší aplikaci	465
29.7 Specifikace jednotlivých účastníků	466
29.8 Společné API	466
29.9 Schopnosti jednotlivých účastníků	467
29.9.1 IGame	467
29.9.2 IWorld	468
29.9.3 INamed	468
29.9.4 IItemContainer	469
29.9.5 IPlace	469
29.9.6 IItem	469
29.9.7 IBag	469
29.9.8 IAction	470

29.10 Shrnutí a soubory pro opakování	470
Kapitola 30 Testovací scénáře – dokončení API	471
30.1 Jak testovat	471
30.1.1 Programování řízené testy	471
30.1.2 Jednotkové, integrační a regresní testy	472
30.1.3 Možnosti testování naší hry	473
30.2 Scénáře	473
30.3 Kroky scénáře – třída ScenarioStep	474
30.4 Výčtový typ TypeOfStep	476
30.5 Výčtový typ TypeOfScenario	476
30.6 Třída Scenario	477
30.7 Postup vývoje a požadavky na scénáře	477
30.7.1 Jednotlivé etapy vývoje	478
30.8 Doplnění API	479
30.8.1 Interfejs IAuthor	479
30.8.2 Interfejs IPortal	479
30.8.3 Třída BasicActions	480
30.8.4 Balíček game77.testers	481
30.8.5 Interfejs IGame	481
30.9 Shrnutí a soubory pro opakování	481
Kapitola 31 První a druhá etapa: scénáře HAPPY a BASIC	482
31.1 Balíček etapy	482
31.2 Návrh portálu	482
31.3 Správce scénářů – ScenarioManager	485
31.4 Test na hladině Level.HAPPY	488
31.5 Druhá etapa – Scénář BASIC	490
31.5.1 Balíček etapy	490
31.5.2 Návrh portálu	490
31.5.3 Úpravy třídy game77.ck1b_duplet.ScenarioManager	490
31.5.4 Test na hladině Level.DUPLET	492
31.6 Shrnutí a soubory pro opakování	493
Kapitola 32 Třetí etapa: Implementace navržené architektury	494
32.1 Tvorba kostry vyhovující API	494
32.2 Návrh tříd v balíčku ck1c_architecture	495
32.2.1 Třída Portal	495
32.2.2 Třída ANamed	495
32.2.3 Třída Action	496
32.2.4 Třída Item	497
32.2.5 Třída AItemContainer	497
32.2.6 Třída Bag	498
32.2.7 Třída Place	499
32.2.8 Třída World	499
32.2.9 Třída Game	501
32.3 Test na hladině Level.ARCHITECTURE	503
32.4 Shrnutí a soubory pro opakování	504
Kapitola 33 Čtvrtá etapa: spuštění hry	505
33.1 Návrh tříd v balíčku ck1d_start	505
33.2 Tři druhy objektů	505
33.3 Delegování zodpovědnosti	506
33.4 Statické metody třídy Action	507
33.4.1 Metody isActive() a stop()	507

33.4.2	Statická metoda <code>executeCommand(String)</code>	507
33.4.3	Definice má být krátká	508
33.4.4	Pomocné statické metody	509
33.5	Úpravy třídy <code>SceneManager</code>	509
33.5.1	Problematika statických konstant	510
33.5.2	Vlastní úprava	510
33.6	Spuštění testu na hladině <code>Level.START</code>	512
33.7	Shrnutí a soubory pro opakování	513
Kapitola 34	Pátá etapa: tvorba světa	514
34.1	Balíček <code>ck1e_world</code> a třída <code>Portal</code>	514
34.2	Vytváříme prostory	514
34.2.1	Inicializace prostoru	515
34.3	Modifikace třídy <code>ItemContainer</code>	516
34.4	Další úpravy třídy <code>SceneManager</code>	517
34.4.1	Simulace chodu hry podle scénáře	519
34.5	Modifikace třídy <code>World</code>	519
34.5.1	Hledání chyby	520
34.6	Test na hladině <code>Level.WORLD</code>	521
34.7	Shrnutí a soubory pro opakování	522
Kapitola 35	Šestá etapa: definice standardních akcí	523
35.1	Balíček <code>ck1f_basic</code> a třída <code>Portal</code>	523
35.2	Definice akcí	523
35.3	Funkce <code>executeStandardCommand(String)</code>	524
35.4	Definice výkonného kódu jednotlivých akcí	525
35.4.1	Akce <code>Jdi</code> a metoda <code>move(String[])</code>	526
35.4.2	Akce <code>Vezmi</code> a metoda <code>take(String[])</code>	527
35.4.3	Akce <code>Polož</code> a metoda <code>put(String[])</code>	528
35.4.4	Akce <code>?</code> a metoda <code>help(String[])</code>	528
35.4.5	Kontrolní test	529
35.5	Shrnutí a soubory pro opakování	529
Kapitola 36	Sedmá a osmá etapa: děláme aplikaci robustní	530
36.1	Nesplněné body zadání	530
36.2	Balíček pro hladinu <code>Level.TRIPLET</code>	531
36.2.1	Proč samostatná etapa	531
36.3	Chybový scénář	531
36.3.1	Co vše se má zkontrolovat	532
36.3.2	Reakce na pokus o nekorektní spuštění	533
36.3.3	Vlastní definice chybového scénáře	533
36.3.4	Test chybového scénáře	533
36.4	Přechod na hladinu <code>Level.MISTAKES</code>	534
36.4.1	Chybné spuštění hry	534
36.5	Problémy s argumenty standardních akcí	535
36.5.1	Nezadané argumenty	535
36.5.2	Finalizace změny prostoru	536
36.5.3	Problémy se zvedáním h-objektu	536
36.6	Změny v definici h-objektů a práce s nimi	537
36.6.1	Nová definice třídy <code>Item</code>	537
36.6.2	Úprava definice prostorů	538
36.6.3	Úprava batohu	539
36.6.4	Dokončení metody <code>take(String[])</code>	539
36.6.5	Dotažení metody <code>put(String[])</code>	541
36.7	Shrnutí a soubory pro opakování	541

Kapitola 37 Etapy 9–11: přidáváme nestandardní akce	542
37.1 Předehra	542
37.1.1 Plánovaný postup –etapy dalšího vývoje.....	542
37.2 RUNNING: Rozchození scénáře HAPPY	543
37.2.1 Úprava scénáře HAPPY.....	543
37.2.2 Úprava konstruktoru akcí	543
37.2.3 Výchozí podoba definic nestandardních akcí.....	545
37.3 QUADRUPLER: Příprava testu robustnosti	545
37.3.1 Nutné podmínky pro korektní zadání nestandardních akcí.....	546
37.3.2 Zanesení nutných podmínek	546
37.3.3 Zanesení nutných podmínek do scénářů	547
37.3.4 Průběžný test	550
37.3.5 Jaká chybná zadání se budou testovat	550
37.3.6 Přidání scénáře MISTAKE_NS	550
37.4 WHOLE: Rozchození výsledného kódu hry	550
37.4.1 Definice metod <code>conditions()</code> a <code>tests()</code> ve třídě <code>Game</code>	552
37.4.2 Definice atributů <code>CONDITIONS</code> a <code>TESTS</code> ve třídě <code>Action</code>	552
37.4.3 Definice kódu nestandardních akcí	553
37.5 Shrnutí a soubory pro opakování	554
Kapitola 38 Závěr: doplnění uživatelského rozhraní	555
38.1 Hlavní třída aplikace	555
38.2 Jednoduché textové uživatelské rozhraní.....	555
38.2.1 Možnost opakovaného spouštění	557
38.2.2 Přidání informací o aktuálním stavu	558
38.3 Opět trocha teorie	558
38.3.1 Vzor Model-View-Controller (MVC)	558
38.3.2 Kompaktnější varianta: M(V+C).....	559
38.4 Volitelné uživatelské rozhraní	559
38.4.1 Interfejs <code>IUI</code>	559
38.4.2 Definice třídy <code>Main2</code>	560
38.4.3 Úprava metod <code>run(...)</code> a <code>multirun(...)</code>	561
38.4.4 Definice třídy <code>Console</code>	563
38.5 Vytvoření primitivního GUI	563
38.5.1 Modalita dialogových oken.....	563
38.5.2 Třída <code>javax.swing.JOptionPane</code>	564
Návrhový vzor Fasáda.....	564
38.5.3 Třída <code>PrimitiveGUI</code>	565
38.6 Několik dalších námětů	566
38.7 Shrnutí a soubory pro opakování	567
Literatura.....	568
Rejstřík	571

Poděkování

Vím, že se v českých knížkách většinou neděkuje, ale tvorba knih je spojena s takovými oběťmi řady lidí z mého blízkého i vzdálenějšího okolí, že bych měl velkou újmu na duši, kdybych tak neučinil.

Chtěl bych především nesmírně poděkovat své ženě Jarušce, která byla po celou dobu mojí největší oporou a jejíž nekonečná trpělivost a vstřícnost mi pomohla dokončit knihu v termínu, který se příliš nelišil od toho, jež jsme původně s nakladatelem dohodli, a ne až někdy za rok po něm. Stále marně přemýšlím, kde má schovanou svatozář.

Původně jsem se domníval, že s dalším vydáním nebude moc práce. Šeredně jsem se zmýlil, protože veškeré úpravy, jež jsem do knihy zanesl, jsou vykoupeny hodinami studia a experimentování, které rodina s neuvěřitelnou trpělivostí snášela.

Na vylepšování textu nového vydání se podílela řada dalších lidí. Mezi nimi musím poděkovat především těm, kteří si dali práci s odhalováním případných chyb ve vznikajícím rukopisu. Mezi nimi pak především Ludkovi Šťastnému, který po celou přípravu rukopis pročetl a odhaloval v něm pasáže, jež by si zasloužily vylepšit.

Neméně velkou zásluhu na současné podobě má i Marek Chadim a Jakub Kolář, kteří se na text dívali očima vedoucích programátorských týmů a doporučili řadu užitečných vylepšení. Řadou podnětných myšlenek přispěl Michal Palas, jenž pracuje v oblasti analýzy a zpracování dat a přispěl tak řadou poznatků z praxe.

V neposlední řadě patří můj dík redakci, především redaktoru Petru Somogyimu, který trpělivě snášel mé neustálé modifikace již zkorigovaného textu, a Radku Matulíkovi, který mne k napsání jednotlivých knih z posledních let vyhecoval a byl pak ochoten pár týdnů počkat, když se mi nepodařilo přesně dodržet původně dohodnutý termín.

Vážení čtenáři, právě jste dočetli ukázkou z knihy ***Programování v jazyku Java***.
Pokud se Vám ukázka líbila, na našem webu si můžete zakoupit celou knihu.