

Obsah

Úvod	11
Jak tato kniha vznikla	11
Co můžete od knížky očekávat	12
Jak jsou organizovány programy	13
Poděkování	14
Kde hledat nejnovější informace	14
Poznámky ke druhému vydání	14
Typografické a syntaktické konvence	16
 1. Pole	 17
1.1. Podpora práce s poli – třída <code>Arrays</code>	17
1.1.1. Základní metody třídy <code>Arrays</code>	18
1.1.2. Práce s vícerozměrnými poli	20
1.2. Řazení objektů	21
1.2.1. Přirozené řazení	21
1.2.2. Absolutní řazení	25
1.2.3. Řazení akcentovaných řetězců	29
1.2.4. Ideální porovnatelná třída	31
1.3. Binární vyhledávání	33
1.4. Práce s částí pole	35
1.5. Méně časté operace s poli	37
1.5.1. Kopírování pole – <code>System.arraycopy()</code>	37
1.5.2. Metoda <code>asList()</code>	40
1.5.3. Metoda <code>equals()</code>	40

2. Kolekce (kontejnery)	45
2.1. Hlavní výhody používání kolekcí	45
2.2. Základní typy kolekcí	46
2.3. Úvod do genericity	50
2.3.1. Použití zástupného znaku v deklaracích	51
2.3.2. Použití zástupného znaku s omezením	52
2.4. Automatické zapouzdřování primitivních datových typů	54
3. Společné metody seznamů a množin	56
3.1. Rozhraní <code>Collection</code>	56
3.1.1. Metody pro plnění kolekce	56
3.1.2. Metody pro ubírání z kolekce	56
3.1.3. Dynamické vlastnosti kolekcí	57
3.1.4. Získání přístupového objektu	57
3.1.5. Převod kolekce na běžné pole	57
3.2. Rozhraní <code>List</code>	57
3.2.1. Změny v kolekci	57
3.2.2. Získání obsahu kolekce	58
3.3. Rozhraní <code>Set</code>	58
4. Třída <code>ArrayList</code>	59
4.1. Velikost versus kapacita	59
4.2. Základní dovednosti s <code>ArrayList</code>	61
4.2.1. Metody z rozhraní <code>Collection</code>	61
4.2.2. Metody z rozhraní <code>List</code> – využití indexů	63
4.3. Vlastní třída jako prvek seznamu	64
4.4. Problémy při práci s podseznamy	66
4.5. Převod seznamu nebo množiny na běžné pole objektů	68
4.6. Převod běžného pole na kolekci	70
4.6.1. Seznam pevné velikosti	70
5. Třída <code>Collections</code>	72
5.1. Nejčastěji používané metody	72
5.1.1. Plnění seznamu	72
5.1.2. Řazení seznamu	72
5.1.3. Binární vyhledávání v seřazeném seznamu	73
5.1.4. Hledání mezních hodnot	73

5.1.5. Otočení pořadí seznamu	74
5.1.6. Zamíchání seznamu	74
5.1.7. Četnost prvků v seznamu	74
5.2. Méně známé metody z Collections	76
5.2.1. Hledání podseznamu	76
5.2.2. Náhrada prvků seznamu	77
5.2.3. Rotace seznamu	77
5.2.4. Výměna prvků	78
6. Speciální druhy seznamů	79
6.1. Spojový seznam – LinkedList	79
6.1.1. Datová struktura zásobník	80
7. Fronta - Queue	83
7.1. Přehled metod	83
7.1.1. Metody pro vkládání	83
7.1.2. Výběr objektu z čela fronty	84
7.1.3. Prohlédnutí objektu na čele fronty	84
7.1.4. Další použitelné metody	84
7.2. Implementace fronty	84
7.2.1. Fronta pomocí PriorityQueue	85
7.2.2. Fronta pomocí LinkedList	87
8. Iterátory	89
8.1. Záměna kolekcí	89
8.2. Iterátor for-each	90
8.3. Klasický iterátor	92
8.4. Třída ListIterator	95
9. Výhodnost jednotlivých seznamů	97
9.1. Srovnání seznamů navzájem	97
9.2. Srovnání seznamů a polí	105
10. Množiny	106
10.1. Rozhraní Set	106
10.2. Implementace množin	107
10.3. Základní dovednosti s množinou	108

10.4. Práce s vlastní třídou v množině	109
10.4.1. Pokusy s překrýváním metod	109
10.5. Množinové operace	112
10.5.1. Triky za použití množinových operací	114
10.6. Použití Collections v množinách	114
10.7. Rozhraní SortedSet a třída TreeSet	116
10.7.1. Použití absolutního řazení u TreeSet	118
10.7.2. Vztah množiny a podmnožiny	118
10.7.3. Triky při použití podmnožin typu TreeSet	119
10.8. Rychlostní parametry	122
10.8.1. Hešovací množina versus seřazená množina	122
10.8.2. Množiny versus seznamy	123
11. Mapy	125
11.1. Implementace map	125
11.2. Rozhraní Map	126
11.3. Základní operace s mapou	128
11.4. Průchod prvky mapy	130
11.5. Změna mapy pomocí pohledů	133
11.6. Triky s mapami	134
11.6.1. Porovnání map	135
11.6.2. Praktické použití map	136
11.7. Třída TreeMap	139
12. Speciální dovednosti s kolekcemi	143
12.1. Vestavěná ochrana kolekcí proti nekonzistenci dat	143
12.2. Přidaná funkčnost – wrapery	145
12.2.1. Synchronizační wrapper	145
12.2.2. Neměnitelný wrapper	147
12.3. Práce se singletony	149
12.4. Možnosti zrychlování práce s hešovací tabulkou	149
13. Speciální typy kolekcí	151
13.1. Třída LinkedHashMap	151
13.2. Třída LinkedHashMap	152

13.3. Třída <code>WeakHashMap</code>	154
13.3.1. Principy „slabých“ referencí	154
13.3.2. Ukázka možností <code>WeakHashMap</code>	155
13.3.3. Praktické použití <code>WeakHashMap</code>	157
13.4. Třída <code>IdentityHashMap</code>	160
14. Jak připravit ideální třídu pro uložení do kolekcí	161
14.1. Hešování	161
14.1.1. Pravidla pro použití hešovacích tabulek	162
14.1.2. Chybné hešování	163
14.2. Obecné principy metody <code>equals()</code>	167
14.2.1. Perfektní <code>equals()</code>	168
14.3. Způsoby přípravy <code>hashCode()</code>	170
14.3.1. Primitivní řešení	170
14.3.2. Hešovací kód může být stejný	171
14.3.3. Hešovací kód se nesmí měnit	173
14.3.4. Doporučení pro přípravu perfektní metody <code>hashCode()</code> ..	174
14.3.5. Praktické použití perfektní <code>hashCode()</code>	176
14.4. Co všechno by měl mít prvek kolekce	181
14.4.1. Typický prvek kolekce pro české řazení	185
15. Extrémně velká celá čísla	191
16. Speciální požadavky na reálné číslo	193
16.1. Vytvoření objektu <code>BigDecimal</code>	194
16.2. Praktické použití <code>BigDecimal</code>	195
16.2.1. Pro přesné výpočty s desetinnými čísly pro hodnoty „běžných“ velikostí	195
16.2.2. Pro přesné výpočty s čísly s extrémní přesností	197
16.2.3. Zaokrouhlování	198
17. Práce s náhodnými čísly	203
17.1. Třída <code>Random</code> – základní použití	203
17.2. Generování čísel z daného intervalu	205
17.3. Normální rozložení pravděpodobnosti	206
17.4. Speciální rozložení pravděpodobnosti	209
17.4.1. Obecné normální rozložení	209
17.4.2. Exponenciální rozložení	211

18. Práce s datem a časem	212
18.1. Třída Date	212
18.2. Třída Calendar	213
18.2.1. Atribut lenient	214
18.2.2. Hraniční časy	214
18.2.3. Konstanty třídy Calendar	215
18.2.4. Získávání hodnot	216
18.2.5. Formátování hodnot pro výpis	218
18.2.5.1. Nevhodné formátování hodnot	218
18.2.5.2. Doporučené formátování hodnot	219
18.2.5.3. Přehled základních formátovacích vzorů	220
18.2.5.4. Složené formátování	221
18.2.6. Získávání speciálních hodnot	222
18.2.7. Nastavení hodnot	223
18.2.8. Přidání hodnoty s kumulativní změnou	225
18.2.9. Přidání hodnoty bez kumulativní změny	226
18.3. Třída GregorianCalendar	228
18.4. Třída TimeZone	230
18.5. Třída SimpleTimeZone	231
19. Zip – komprimace dat	234
19.1. Třída ZipFile	234
19.2. Třída ZipInputStream	235
19.3. Třída ZipEntry	236
19.3.1. Použití ZipEntry pro čtení	236
19.3.2. Použití ZipEntry pro vytvoření komprimovaných dat ..	240
19.4. Třída ZipOutputStream – zápis komprimovaných dat	241
Literatura	244
Rejstřík	245