

Stručný obsah

| | |
|--|------------|
| Poděkování | 18 |
| Předmluva | 19 |
| Úvod | 20 |
| Část 1: Zapouzdření | 27 |
| 1. Seznamujeme se s nástroji | 28 |
| 2. Pracujeme s třídami a objekty | 50 |
| 3. Vytváříme vlastní třídu | 90 |
| 4. Dotváříme vlastní třídu | 192 |
| 5. Návrhové vzory | 244 |
| Část 2: Více tváří | 259 |
| 6. Rozhraní | 260 |
| 7. Budete si to přát zabalit? | 310 |
| 8. Co takhle něco zdědit? | 335 |
| 9. Dědit mohou i třídy | 354 |
| 10. Knihovny | 430 |
| Část 3: Učíme program přemýšlet | 449 |
| 11. Program začíná přemýšlet | 450 |
| 12. Ještě jednu rundu, prosím | 494 |
| 13. Kontejnery nejsou jen na odpadky | 515 |
| 14. Statické kontejnery – pole | 563 |
| 15. Závěrečný projekt a kudy dál | 587 |
| Rejstřík | 594 |

První vydání: Praha 2004

Vydání 78. letk číslo

Podkapitola 78.Druhy

ISBN 978-80-247-2941-4

Podrobný obsah

| | |
|--|-----------|
| Poděkování | 18 |
| Předmluva | 19 |
| Úvod | 20 |
| Potřebné vybavení | 23 |
| Použité konvence | 24 |
| Odbočka | 26 |
| Část 1: Zapouzdření | 27 |
| 1. Seznamujeme se s nástroji | 28 |
| 1.1 Trochu historie | 28 |
| První počítače | 28 |
| Co je to program | 30 |
| Program musí být především spolehlivý | 30 |
| 1.2 Objektově orientované programování – OOP | 31 |
| Vývoj metodik programování | 31 |
| Principy OOP | 32 |
| 1.3 Překladače, interprety, platformy | 33 |
| Operační systém a platforma | 33 |
| Programovací jazyky | 34 |
| 1.4 Java a její zvláštnosti | 36 |
| Klíčové vlastnosti Javy | 36 |
| Objektově orientovaná | 36 |
| Jednoduchá | 36 |
| Multiplatformní | 37 |
| Java je jazyk i platforma | 37 |
| Vývojářská sada | 38 |
| 1.5 Vývojové prostředí BlueJ | 38 |
| 1.6 Projekty a BlueJ | 39 |
| Windows a substituované disky | 40 |
| Umístění projektů na disku | 41 |
| Vyhledání a otevření projektu | 42 |
| 1.7 Diagram tříd | 43 |
| Manipulace s třídami v diagramu | 44 |
| 1.8 Shrnutí – co jsme se naučili | 48 |
| 2. Pracujeme s třídami a objekty | 50 |
| 2.1 Nejprve trocha teorie | 50 |
| Třídy a jejich instance | 50 |
| Zprávy | 51 |
| Metody | 52 |

| | | |
|-------|--|-----|
| 12. | Ještě jednu rundu, prosím | 494 |
| 12.1 | Cykly | 494 |
| 12.2 | Jak máme rychlý počítač – cyklus s koncovou podmínkou | 495 |
| 12.3 | Jeden test nestačí – cyklus s počáteční podmínkou | 496 |
| 12.4 | Cyklus s parametrem | 497 |
| 12.5 | Nekonečný cyklus | 499 |
| 12.6 | Vnořování cyklů | 500 |
| 12.7 | Cyklus s podmínkou uprostřed | 501 |
| 12.8 | Cyklus s prázdným tělem | 502 |
| 12.9 | Skákající balónek | 503 |
| | Zadání | 503 |
| | Příprava testu | 503 |
| | Předběžné úvahy, definice konstruktorů | 504 |
| | Koncepte simulace pádu | 505 |
| | Dotážení simulace pádu | 506 |
| | Metody přemístění <code>Na(int,int)</code> a <code>spadni()</code> | 508 |
| | Balón se odráží | 508 |
| | Zmenšování odrazů | 509 |
| 12.10 | Jak dělat několik věcí najednou | 510 |
| | Vlákna | 511 |
| | Spuštění pádu v samostatném vlákně | 511 |
| 12.11 | Shrnutí – co jsme se naučili | 513 |
| 13. | Kontejnery nejsou jen na odpadky | 515 |
| 13.1 | Co je to kontejner | 516 |
| | Množina (Set) | 516 |
| | Další kontejnery | 517 |
| | Zásobník (Stack) | 517 |
| | Fronta (Queue) | 517 |
| | Strom (Tree) | 517 |
| | Graf | 517 |
| | Mapa (Map), Slovník (Dictionary) | 518 |
| | Seznam (List) | 518 |
| 13.2 | Koncepte kontejnerů ve standardní knihovně | 518 |
| | Deklarujte typy co nejobecněji | 519 |
| | Zadání typu prvků obsažených v kontejneru | 520 |
| | Rozhraní <code>java.util.Collection<E></code> | 521 |
| 13.3 | Pracujeme s množinami | 522 |
| | Rozhraní <code>java.util.Set<E></code> | 522 |
| | Třída <code>java.util.LinkedHashSet<E></code> | 522 |
| 13.4 | Brownův pohyb molekul | 523 |
| | 1. Konstrukce molekuly | 524 |
| | 2. Náhodné rozmístění molekul | 525 |
| | 3. Pohyb molekul a jejich srážky | 528 |
| | 4. Animátor | 529 |
| | Pravidelné spuštění úloh pomocí instance třídy <code>java.util.Timer</code> | 530 |
| 13.5 | Soukromé třídy | 532 |
| 13.6 | Návrhový vzor Iterátor (Iterator) | 533 |
| | Rozhraní <code>java.util.Iterátor<E></code> | 534 |
| | Molekuly s výlevkou | 535 |
| 13.7 | Pracujeme se seznamy | 538 |
| | Rozhraní <code>java.util.List<E></code> | 538 |
| | Třídy <code>java.util.ArrayList<E></code> a <code>java.util.LinkedList<E></code> | 539 |

| | | |
|---------------|---|-----|
| 13.8 | Mnohotvar..... | 539 |
| | Základní koncepce a první testy..... | 540 |
| | Dovedení programu k úspěšnému vykonání testů..... | 543 |
| | Konstruktory..... | 543 |
| | Metoda nakresli(Kreslítko)..... | 544 |
| | Metoda přidej(IHýbaci)..... | 544 |
| | Přidání hýbacích vlastností..... | 546 |
| | Metoda setPozice(int,int)..... | 547 |
| | Metoda setRozměr(int,int)..... | 548 |
| 13.9 | Soukromá přepravka..... | 550 |
| 13.10 | Zavedení vrstev – práce se seznamy..... | 556 |
| | Třída java.util.ListIterator<E>..... | 559 |
| 13.11 | Primitivní a obalové datové typy..... | 560 |
| 13.12 | Shrnutí – co jsme se naučili..... | 561 |
| 14. | Statické kontejnery – pole..... | 563 |
| 14.1 | Pole jako kontejner..... | 564 |
| | Pole odkazů na objekty..... | 564 |
| | Pole a BlueJ..... | 565 |
| | Pole hodnot primitivních typů..... | 567 |
| | Hlídaní mezi poli..... | 570 |
| | Inicializace polí v deklaraci..... | 570 |
| | Inicializace vytvářeného pole..... | 572 |
| | Neinicializovaná pole objektových typů..... | 573 |
| 14.2 | Vypsání čísla slovy..... | 574 |
| 14.3 | Vícerozměrná pole..... | 576 |
| | Obdélníková pole..... | 576 |
| | Neobdélníková pole..... | 578 |
| | Inicializace vícerozměrného pole..... | 578 |
| 14.4 | Pascalův trojúhelník..... | 579 |
| 14.5 | Třídy StringBuilder a StringBuffer..... | 580 |
| 14.6 | Metoda main(String[])..... | 581 |
| 14.7 | Metody s proměnlivým počtem parametrů..... | 582 |
| 14.8 | Shrnutí – co jsme se naučili..... | 584 |
| 15. | Závěrečný projekt a kudy dál..... | 587 |
| 15.1 | Závěrečný projekt: Displej..... | 587 |
| | Zadání..... | 587 |
| | Analýza..... | 588 |
| | Displej..... | 589 |
| | Číslice..... | 589 |
| | Segment..... | 589 |
| | Zpět u číslic..... | 590 |
| | Dotahujeme segmenty..... | 590 |
| | Dotahujeme číslice..... | 591 |
| | Dotahujeme displej..... | 592 |
| | Závěr..... | 592 |
| 15.2 | Kudy dál..... | 592 |
| Rejstřík..... | | 594 |

| | | |
|------|--|-----|
| 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 | 60 |
| | Rušení instancí a správa paměti | 61 |
| 2.4 | Restartování virtuálního stroje | 62 |
| 2.5 | Instance versus odkaz | 62 |
| 2.6 | Zprávy žádající o hodnotu | 64 |
| | Datové typy | 65 |
| | Primitivní datové typy | 65 |
| | Objektové datové typy | 66 |
| | Vracení hodnot primitivních typů | 67 |
| | Vracení hodnot objektových typů | 68 |
| 2.7 | Parametry a jejich typy | 71 |
| | Vyvolání konstruktoru s parametry | 71 |
| | Parametry objektových typů | 74 |
| | Posílání zpráv s parametry | 75 |
| 2.8 | Metody třídy | 76 |
| 2.9 | Výlet do nitra instancí | 78 |
| | Atributy instancí | 79 |
| | Atributy třídy – statické atributy | 81 |
| 2.10 | Přímé zadávání hodnot parametrů objektových typů | 84 |
| | Veřejné atributy | 84 |
| | Odkazy vrácené po zaslání zprávy | 85 |
| 2.11 | Shrnutí – co jsme se naučili | 87 |
| 3. | Vytváříme vlastní třídu | 90 |
| 3.1 | První vlastní třída | 91 |
| 3.2 | Zdrojový kód třídy | 94 |
| | Prázdná třída | 95 |
| | Bílé znaky a uspořádání programu | 96 |
| | Implicitní konstruktor | 97 |
| 3.3 | Odstranění třídy | 98 |
| 3.4 | Bezparametrický konstruktor | 98 |
| 3.5 | Přejmenování třídy | 102 |
| 3.6 | Ladění | 103 |
| | Syntaktické chyby | 104 |
| | Běhové chyby | 106 |
| | Logické (sémantické) chyby | 107 |
| 3.7 | Konstruktor s parametry | 108 |
| | Konstruktor this | 110 |
| 3.8 | Přetěžování | 113 |
| 3.9 | Testování | 114 |
| | TDD – vývoj řízený testy | 114 |
| | Zprovoznění nástrojů pro automatizaci testů | 115 |
| | Testovací třída | 116 |
| | Přípravek | 117 |
| | Úprava obsahu přípravku | 118 |
| 3.10 | Deklarace atributů | 120 |
| | Modifikátory přístupu | 121 |

| | | |
|------|---|-----|
| | Vylepšujeme Strom | 122 |
| | Možné důsledky zveřejnění atributů | 123 |
| 3.11 | Definujeme vlastní metodu | 124 |
| | Test vytvořených metod | 126 |
| | Reakce na chybu v testu | 129 |
| | Nejprve testy, pak program? | 130 |
| | Někdy jsou věci složitější | 132 |
| | Použití metod vracejících hodnotu | 134 |
| | Definice metod vracejících hodnotu | 135 |
| | Parametry a návratové hodnoty objektových typů | 136 |
| 3.12 | Doplnění projektu o třídu odjinud | 137 |
| 3.13 | Přístupové metody | 137 |
| | Atributy versus vlastnosti | 139 |
| | Konvence pro názvy přístupových metod | 139 |
| 3.14 | Kvalifikace a klíčové slovo this | 140 |
| | Kvalifikace metod | 140 |
| | Kvalifikace atributů | 142 |
| 3.15 | Atributy a metody třídy (statické atributy a metody) | 144 |
| | Atributy třídy | 144 |
| | Metody třídy | 145 |
| 3.16 | Lokální proměnné | 147 |
| 3.17 | Konstanty a literály | 150 |
| | Konstanty objektových typů | 153 |
| | Správná podoba literálů | 153 |
| | boolean | 154 |
| | int | 154 |
| | double | 154 |
| | String | 155 |
| | null | 156 |
| 3.18 | Zapouzdření | 156 |
| | Rozhraní versus implementace | 156 |
| | Kontrakt | 158 |
| 3.19 | Komentáře a dokumentace | 159 |
| | Proč psát srozumitelné programy | 159 |
| | Tři druhy komentářů | 160 |
| | Uspořádání jednotlivých prvků v těle třídy | 168 |
| | Zakomentování a odkomentování částí programu | 169 |
| | BlueJ a komentářová nápověda | 170 |
| | Automaticky generovaná dokumentace | 171 |
| | Dokumentace celého projektu | 173 |
| | Pomocné značky pro tvorbu dokumentace | 174 |
| 3.20 | Závěrečný příklad – UFO | 176 |
| | Třída Dispečer | 177 |
| | Jednodušší varianta | 178 |
| | Varianta ovládaná z klávesnice | 178 |
| | Třída UFO | 179 |
| | Atributy | 179 |
| | Konstruktor | 180 |
| | Metoda setRychlost(int,int) | 180 |
| | Metody getX(), getY(), getXRychlost(), getYRychlost(), getXTah(), getYTah() | 180 |
| | Metoda nakresli() | 180 |
| | Metoda popojed(int) | 180 |
| | Metody vpravo(), vlevo(), vzhůru(), dolů(), vypniMotory() | 181 |
| | Třída UFO_3a | 182 |

| | | |
|------|--|-----|
| | Třída UFOTest | 182 |
| 3.21 | Vytvoření samostatné aplikace | 183 |
| | Třída spouštějící aplikaci | 183 |
| | Prohlížení obsahu JAR souborů | 184 |
| | Vytvoření souboru JAR s aplikací | 184 |
| 3.22 | Shrnutí – co jsme se naučili | 186 |
| | Zdrojový kód | 186 |
| | Ladění | 187 |
| | Konstruktory a metody | 188 |
| | Atributy a lokální proměnné | 189 |
| | Dokumentace | 190 |
| | Aplikace | 191 |
| 4. | Dotváříme vlastní třídu | 192 |
| 4.1 | Jednoduché vstupy a výstupy | 192 |
| | Textové řetězce | 193 |
| | Rozdíl mezi prázdným řetězcem a null | 195 |
| | Čísla | 195 |
| 4.2 | Knihovny statických metod | 197 |
| 4.3 | Podrobnosti o operátorech | 197 |
| | Binární aritmetické operátory + - * / % | 198 |
| | Sčítání, odčítání, násobení | 198 |
| | Slučování řetězců + | 199 |
| | Dělení / | 199 |
| | Zbytek po dělení (dělení modulo) % | 200 |
| | Unární operátory + - | 201 |
| | Kulaté závorky () | 201 |
| | Přířazovací operátor = | 201 |
| | Složené přířazovací operátory +=, -=, *=, /=, %= | 202 |
| | Operátor přetypování (typ) | 203 |
| | Univerzální přetypování na String | 204 |
| 4.4 | Počítáme instance | 205 |
| 4.5 | Inkrementační a dekrementační operátory | 206 |
| | Způsoby předávání hodnot | 209 |
| | Jiný způsob inicializace rodného čísla | 210 |
| 4.6 | Standardní výstupy | 211 |
| | Standardní chybový výstup | 213 |
| 4.7 | Metoda toString() | 214 |
| 4.8 | Prázdná standardní třída | 215 |
| 4.9 | V útrokách testovací třídy | 218 |
| | Přípravek | 220 |
| | Automaticky generované testy | 222 |
| | Vlastní testy | 223 |
| | Úklid | 224 |
| | Metody assertEquals a assertTrue | 224 |
| | Test testů | 225 |
| 4.10 | Debugger a práce s ním | 227 |
| | Krokování programu | 228 |
| | Okno debuggeru | 231 |
| | Vlákna | 232 |
| | Pořadí volání | 232 |
| | Atributy třídy | 233 |
| | Atributy instancí | 233 |
| | Lokální proměnné | 233 |

| | | |
|------|--|-----|
| | Atributy a proměnné objektových typů | 234 |
| | Už nezastavuj – ruším zářky..... | 235 |
| | Předčasný konec programu..... | 236 |
| | Pozastavení běžícího programu | 236 |
| | Krokování konstruktorů | 237 |
| 4.11 | Hodnotové a referenční typy | 237 |
| | Hodnotové typy | 237 |
| | Referenční datové typy..... | 238 |
| | Program demonstující rozdíl | 238 |
| 4.12 | Projekt Zlomky | 240 |
| 4.13 | Shrnutí – co jsme se naučili | 241 |
| 5. | Návrhové vzory..... | 244 |
| 5.1 | Přepravka (Messenger)..... | 245 |
| 5.2 | Tovární metoda (Factory method)..... | 249 |
| 5.3 | Jedináček (Singleton) | 250 |
| 5.4 | Výčtové typy | 253 |
| 5.5 | Shrnutí – co jsme se naučili | 257 |

Část 2: Více tváří 259

| | | |
|-----|---|-----|
| 6. | Rozhraní | 260 |
| 6.1 | Kreslíme jinak | 261 |
| 6.2 | Syntaxe rozhraní | 262 |
| 6.3 | Instance rozhraní..... | 264 |
| 6.4 | Nový projekt | 265 |
| | Práce s novým plátnem | 268 |
| | Událostmi řízené programování | 270 |
| 6.5 | Implementace rozhraní | 270 |
| | Implementace rozhraní v diagramu tříd | 271 |
| | Odvolání implementace rozhraní..... | 272 |
| | Implementace rozhraní ve zdrojovém kódu | 272 |
| 6.6 | Úprava zdrojového kódu třídy Strom | 273 |
| | Třída musí jít přeložit | 274 |
| | Testování..... | 277 |
| | Opomenuté testy | 281 |
| | Závěrečné úpravy..... | 282 |
| | Uložení odkazu na Plátno do atributu třídy | 282 |
| | Odstranění statického atributu krok | 282 |
| | Úpravy posunových metod..... | 283 |
| | Zefektivnění přesunu..... | 283 |
| | Efektivita vykreslování | 284 |
| 6.7 | Implementace několika rozhraní..... | 285 |
| 6.8 | Návrhový vzor Služebník (Servant) | 286 |
| | Proč rozhraní | 287 |
| | Implementace | 287 |
| | Aplikace na náš projekt..... | 288 |
| | Závěrečný test | 290 |
| 6.9 | Refaktorování | 291 |
| | Ukázka | 293 |
| | 1. krok: Vytvoření testu | 293 |
| | 2. krok: Doplnění prázdných verzí testovaných metod | 294 |
| | 3. krok: Definice nových atributů..... | 294 |

| | | |
|-------------|---|------------|
| | 4. krok: Kopírování těla konstrukturu do těla metody | 295 |
| | 5. krok: Dočasné „odkonstantnění“ některých atributů | 296 |
| | 6. krok: Definice potřebných lokálních proměnných | 296 |
| | 7. krok: Odstranění tvorby nových instancí koruny a kmene | 297 |
| | 8. krok: Vrácení koruny a kmene mezi konstanty | 297 |
| | 9. krok: Vyvolání metody setRozměr(int,int) v konstrukturu | 297 |
| | 10. krok: Odstranění zdvojeného kódu z konstrukturu | 298 |
| | 11. krok: Doplnění metody setRozměr(Rozměr) | 299 |
| | 12. krok: Doplnění metody setOblast(Oblast) | 299 |
| 6.10 | Projekt Výtah | 300 |
| | Analýza problému | 300 |
| | Okolí | 301 |
| | Konstruktory | 301 |
| | Potřebné metody | 302 |
| | Implementace | 302 |
| | Implementovaná rozhraní | 303 |
| | Atributy | 303 |
| | Postup při návrhu metod | 304 |
| | Metoda doPatra(int) | 304 |
| | Metoda přijedK(IPosuvný) | 305 |
| | Metoda nástup(IPosuvný) | 305 |
| | Metody výstupVpravo() a výstupVlevo() | 306 |
| | Test převozu pasažéra | 306 |
| | Metody odvezVpravo(IPosuvný,int) a odvezVlevo(IPosuvný,int) | 307 |
| 6.11 | Shrnutí – co jsme se naučili | 307 |
| 7. | Budete si to přát zabalit? | 310 |
| 7.1 | Velké programy a jejich problémy | 310 |
| 7.2 | Balíčky | 311 |
| | Podbalíčky | 312 |
| | Uspořádání podbalíčků s programy k dosavadní části knihy | 313 |
| | Názvy tříd | 314 |
| 7.3 | Balíčky a BlueJ | 314 |
| | Příprava stromu balíčků pro BlueJ ve správci souborů | 315 |
| | Příprava stromu balíčků v BlueJ | 315 |
| | Vytvoření struktury balíčků pro tuto kapitolu | 316 |
| | Putování stromem balíčků | 316 |
| | Odstraňování balíčků | 317 |
| | Zavírání a otevírání projektů | 318 |
| 7.4 | Naplnujeme balíčky | 318 |
| | Automatické vložení příkazu package | 320 |
| 7.5 | Balíčky a příkaz import | 321 |
| | Import celého balíčku | 323 |
| | Import a podbalíčky | 324 |
| | Balíček java.lang | 324 |
| | Změna balíčku | 324 |
| 7.6 | Názvy balíčků | 325 |
| 7.7 | Příkazový panel | 325 |
| | Nevýhody koncepce balíčků v BlueJ | 325 |
| | Zobrazení příkazového panelu | 326 |
| | Použití příkazového panelu | 327 |
| | Opakované používání příkazů | 328 |
| 7.8 | Přístupová práva v rámci balíčku | 329 |
| 7.9 | Neveřejné třídy | 330 |

| | | |
|-----------|---|------------|
| 7.10 | Tvorba vlastních aplikací..... | 331 |
| 7.11 | Statický import | 332 |
| 7.12 | Shrnutí – co jsme se naučili | 332 |
| 8. | Co takhle něco zdědit? | 335 |
| 8.1 | Co to je, když rozhraní dědí? | 336 |
| 8.2 | Jak to zařídit | 337 |
| | Duplicitně deklarovaná implementace..... | 339 |
| 8.3 | Společný potomek několika rozhraní..... | 339 |
| 8.4 | Návrhový vzor Stav (State)..... | 341 |
| | Projekt Šípky | 342 |
| | Shrnutí..... | 346 |
| 8.5 | Návrhový vzor Zástupce (Proxy) | 347 |
| 8.6 | Projekt Kabina | 349 |
| | Předpřipravené třídy | 349 |
| | Třída rup.česky.tvary.Multipřesouvač | 349 |
| | Rozhraní rup.česky.tvary.IMultiposuvný | 350 |
| | Rozhraní doprava.IZastávka | 350 |
| | Třída doprava.Linka | 351 |
| | Úloha: třída doprava.Kabina | 352 |
| 8.7 | Shrnutí – co jsme se naučili | 352 |
| 9. | Dědit mohou i třídy..... | 354 |
| 9.1 | Podtřídy a nadtřídy..... | 355 |
| | Specializace..... | 355 |
| | Zobecnění | 355 |
| | Realizace v OOP..... | 356 |
| | Univerzální (pra)rodíč | 357 |
| 9.2 | Experimenty s dědičností..... | 358 |
| | Univerzální rodič Object | 359 |
| | Atributy a bezparametrické konstruktory tříd v projektu | 360 |
| | Hierarchie dědičnosti..... | 361 |
| | Podobjekt rodičovské třídy | 363 |
| | Explicitní volání konstruktoru předka..... | 367 |
| | Chráněné atributy – modifikátor přístupu protected | 370 |
| | Dědičnost a metody tříd | 370 |
| | Metody instancí, jejich dědění a překrývání | 372 |
| | Nové metody | 372 |
| | Nepřekryté zděděné metody | 372 |
| | Překryté zděděné metody | 372 |
| | Test chování překrývajících a překrytých metod..... | 374 |
| | Porovnání | 376 |
| | Podobjekt | 376 |
| | Soukromá metoda | 377 |
| | Veřejná metoda | 377 |
| | Instance vnučka | 377 |
| | Vyvolání překryté verze metody | 377 |
| 9.3 | Vytváříme dceřinou třídu | 379 |
| | Jednoduchá dceřiná třída | 379 |
| | Konstruktory potomka..... | 382 |
| | Složitější dceřiná třída | 382 |
| | Definice konstruktůrů | 383 |
| | Metoda kresli(Kreslitko) | 384 |
| | Metoda setPozice(int,int)..... | 384 |

| | | |
|-------------|--|------------|
| | Jak přesvědčit objekt, aby se pokaždé choval jinak | 387 |
| | Samostatná úloha: Terč | 389 |
| 9.4 | Vytváříme rodičovskou třídu | 392 |
| | Společný rodič Posuvný | 392 |
| | Příprava | 392 |
| | Konstantní atributy třídy | 393 |
| | Proměnné atributy třídy | 394 |
| | Konstantní atributy instancí | 394 |
| | Proměnné atributy instancí | 394 |
| | Konstruktory | 396 |
| | Metody instancí | 396 |
| | Doladění dceřiných tříd | 397 |
| | Elipsa, Obdélník, Trojúhelník | 398 |
| | Čára | 399 |
| | Text | 399 |
| | Strom | 399 |
| | Dodatečné rozšíření rodičovské třídy | 400 |
| | Společný rodič Hýbací | 402 |
| 9.5 | Abstraktní metody a třídy | 403 |
| | Neimplementovaná metoda implementovaného rozhraní | 404 |
| | Zděděná a neimplementovaná abstraktní metoda | 405 |
| | Přidání metody nakreslí() | 406 |
| | Nově deklarovaná abstraktní metoda | 406 |
| | Abstraktní třída bez abstraktních metod | 407 |
| 9.6 | Nová schopnost – přizpůsobivost | 408 |
| 9.7 | Návrhový vzor Stav podruhé | 409 |
| | Projekt Šipka | 409 |
| 9.8 | Co je na dědičnosti špatné | 411 |
| 9.9 | ZpětnáKabina | 412 |
| 9.10 | Omezení kladená na konstruktory | 414 |
| | Poznámka o dobrých mravech | 416 |
| 9.11 | Konečné třídy | 417 |
| 9.12 | Konečné metody | 418 |
| 9.13 | ZpětnáKabina podruhé | 419 |
| 9.14 | Tovární metoda podruhé | 420 |
| | Jak něco udělat před spuštěním rodičovského konstrukturu | 420 |
| | Využití tovární metody | 421 |
| 9.15 | Kdy (ne)použít dědičnost | 423 |
| | Potomci, kteří nejsou speciálním případem rodiče | 423 |
| | Kdy jsme použili dědičnost místo správného skládání | 424 |
| | Potomci, kteří jsou příliš speciální | 425 |
| | Kdy dát přednost skládání a kdy dědičnosti | 426 |
| 9.16 | Shrnutí – co jsme se naučili | 427 |
| 10. | Knihovny | 430 |
| 10.1 | Datové typy char a long | 431 |
| | long | 431 |
| | char | 431 |
| 10.2 | Primitivní a obalové datové typy | 433 |
| 10.3 | Knihovní třída jako návrhový vzor | 434 |
| 10.4 | Třída System | 434 |
| 10.5 | Formátovaný výstup | 435 |
| | Národní prostředí | 437 |
| | Ukázka | 437 |

| | | |
|-------|---|-----|
| 10.6 | Základní matematické funkce | 438 |
| 10.7 | Pracujeme s náhodou | 439 |
| 10.8 | Ukončení aplikace | 441 |
| 10.9 | Třída String | 442 |
| 10.10 | Definice vlastní knihovny a její začlenění do BlueJ | 443 |
| | Vytvoření JAR souboru s knihovnou | 444 |
| | Přidání knihovny do BlueJ | 444 |
| 10.11 | Shrnutí – co jsme se naučili | 446 |

Část 3: Učíme program přemýšlet **449**

| | | |
|------------|---|------------|
| 11. | Program začíná přemýšlet | 450 |
| 11.1 | Ternární operátor ? | 451 |
| 11.2 | Jednoduchý podmíněný příkaz | 451 |
| | Třídy jako objekty | 454 |
| 11.3 | Blok příkazů (složený příkaz) | 455 |
| 11.4 | Program ve výjimečné situaci | 457 |
| | Nejdůležitější výjimky | 458 |
| | Vyhození výjimky | 459 |
| | Výjimky a dostupný kód | 460 |
| | Užitečné metody výjimek | 461 |
| | Vytvoření objektu výjimky | 462 |
| | Metoda printStackTrace() | 462 |
| | Metoda printStackTrace(PrintStream) | 462 |
| | Metody getMessage() a toString() | 462 |
| | Vyhození výjimky | 463 |
| 11.5 | Podmínky a jejich skládání | 463 |
| | Porovnávací operátory | 463 |
| | Logické výrazy | 464 |
| | Použití v programu | 465 |
| 11.6 | Návrhový vzor Adaptér (Adapter) | 466 |
| 11.7 | Ošetření klávesnice | 466 |
| | Možné události klávesnice | 467 |
| | Co prozradí událost java.awt.event.KeyEvent | 468 |
| 11.8 | Střelba | 470 |
| | Střela | 471 |
| | Dělo | 472 |
| 11.9 | Statický konstruktor | 473 |
| | Vylepšené dělo | 475 |
| 11.10 | Rychlost ošetření klávesnice | 478 |
| 11.11 | Vnořené podmíněné příkazy | 478 |
| 11.12 | Výběr ze dvou možností | 480 |
| 11.13 | Kaskáda možností | 482 |
| 11.14 | Přepínač | 484 |
| 11.15 | Sestřelování letadel | 487 |
| 11.16 | Přepínač nad výčtovým typem | 487 |
| 11.17 | Operátor instanceof | 488 |
| 11.18 | Metoda equals() | 489 |
| | Překrytí metody equals(Object) | 490 |
| 11.19 | Shrnutí – co jsme se naučili | 491 |