

Část G

Seznamy

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

Seznam S1

Výpisy programů

Výpis 2.1:	Přepsaný obsah konzolového okna otevřeného po spuštění dávky !_JShell.bat na mém počítači.....	47
Výpis 2.2:	První tři pokusné úryvky.....	48
Výpis 2.3:	Více úryvků na jednom řádku.....	51
Výpis 2.4:	Výpisy úryvků příkazem /list.....	53
Výpis 2.5:	Výpis všech úryvků příkazem /list -all.....	54
Výpis 2.6:	Různé způsoby uložení dosavadní práce příkazem /save.....	55
Výpis 2.7:	Vliv otevření skriptu PRINTING.....	56
Výpis 2.8:	Opětovné načtení předchozího stavu příkazem /reload -restore.....	57
Výpis 3.1:	Komentáře.....	64
Výpis 3.2:	Zadávání celočíselných hodnot.....	65
Výpis 3.3:	Zadávání reálných hodnot.....	66
Výpis 3.4:	Zadávání znaků.....	67
Výpis 3.5:	Literály typu String.....	69
Výpis 3.6:	Textové bloky.....	72
Výpis 3.7:	Logické hodnoty.....	72
Výpis 4.1:	Deklarace proměnných.....	77
Výpis 4.2:	Tři druhy dělení.....	81
Výpis 4.3:	Implicitní a explicitní přetypování.....	83
Výpis 4.4:	Použití složeného přiřazovacího příkazu.....	86
Výpis 4.5:	Inkrementační a dekrementační operátory.....	87
Výpis 4.6:	Porovnávací operátory.....	88
Výpis 5.1:	Práce s atributy.....	98
Výpis 5.2:	Volání metod třídy Math.....	99
Výpis 5.3:	Vytváření nových objektů.....	100
Výpis 6.1:	Import datového typu java.util.Date() a jeho následné použití.....	106
Výpis 6.2:	Zobrazení aktuálně importovaných tříd.....	107
Výpis 6.3:	Zpřístupnění knihovny robota Karla.....	109
Výpis 6.4:	Přidání prohledávané složky při spuštění JShell.....	110
Výpis 6.5:	Vytvoření světa, přidání robota a provedení příkazů.....	112
Výpis 6.6:	Ovládání vytvořeného robota.....	114
Výpis 6.7:	Demonstrace vlivu skrývání.....	115
Výpis 6.8:	Opětná aktivace příkazem /reload -restore.....	117
Výpis 7.1:	Definice metod kDoubleStep().....	122
Výpis 7.2:	Definice a použití metod doubleStep(Karel) a turnAbout(Karel).....	124

Výpis 7.3:	Definice a použití metody <code>turnWithInfo(Karel, String)</code>	126
Výpis 7.4:	Demonstrace použití lokálních konstant	127
Výpis 7.5:	Definice a použití přetížené metody <code>turnWithInfo(String, Karel)</code>	130
Výpis 7.6:	Použití přetížených verzí konstruktorů třídy <code>Karel</code>	132
Výpis 7.7:	Definice a použití metody <code>wallBehind(Karel)</code> vracející hodnotu.....	134
Výpis 7.8:	Přehled doposud definovaných metod.....	135
Výpis 7.9:	Definice a použití metod <code>forwardClosed(Karel)</code> , <code>forwardFree(Karel)</code> a <code>rightFree(Karel)</code> používajících logické výrazy a vracejících hodnotu.....	136
Výpis 7.10:	Demonstrace předávání argumentu odkazem v jazyku <code>C#</code>	138
Výpis 7.11:	Obdobná metoda v jazyku <code>Java</code>	139
Výpis 8.1:	Použití cyklu <code>while</code> na příkladu metod <code>goToWall(Karel)</code> a <code>turnToNorth(Karel)</code>	142
Výpis 8.2:	Použití cyklu <code>do...while</code> na příkladu metod <code>goToEmpty(Karel)</code> a <code>markersToWall(Karel)</code>	144
Výpis 8.3:	Použití cyklu <code>for</code> na příkladu metod <code>putNMarkers(Karel, int)</code> , <code>putIncreasingMarkers(Karel)</code> a <code>markers(Karel)</code>	147
Výpis 8.4:	Použití nekonečného cyklu na příkladu metody <code>around(Karel)</code>	149
Výpis 8.5:	Definice metody <code>clearToWall(Karel)</code>	150
Výpis 9.1:	Použití jednoduchého podmíněného příkazu.....	153
Výpis 9.2:	Použití úplného podmíněného příkazu.....	155
Výpis 9.3:	Použití podmíněného výrazu.....	156
Výpis 9.4:	Zanořené zarovnaný složený podmíněný příkaz	157
Výpis 9.5:	Doporučené formátování při výběru z více možností	157
Výpis 9.6:	Využití složeného podmíněného příkazu při průchodu bludištěm.....	157
Výpis 9.7:	Koncepce přepínače (příkazu <code>switch</code>) převzatá z jazyka <code>C</code>	159
Výpis 9.8:	Novější koncepce přepínače (příkazu <code>switch</code>).....	160
Výpis 9.9:	Použití klasické podoby přepínače.....	162
Výpis 9.10:	Použití přepínacího výrazu.....	163
Výpis 9.11:	Definice metody <code>turnTo(Karel, Direction)</code> s využitím klasické verze přepínače	164
Výpis 9.12:	Definice metody <code>markersToWall(Karel)</code> používající cyklus s podmínkou uprostřed	165
Výpis 9.13:	Definice funkce <code>sum3xC()</code> demonstrující použití příkazu <code>continue</code>	166
Výpis 9.14:	Definice funkce <code>factorial()</code> demonstrující použití rekurze	167
Výpis 9.15:	Definice funkce <code>put_markers_at_wall()</code> řešené pomocí rekurze.....	168
Výpis 10.1:	Nejjednodušší definice třídy a vytvoření její instance	173
Výpis 10.2:	Definice a použití třídy <code>Light</code> se třemi konstruktory.....	178
Výpis 10.3:	Upravená definice třídy <code>Light</code> v souboru <code>s10a_Light.jsh</code>	186
Výpis 10.4:	Prověření funkce třídy <code>Light</code> ze souboru <code>s10a_Light.jsh</code>	187
Výpis 11.1:	Ekvivalent třídy <code>Light</code> , která je řešením úlohy 10.1 na straně 187, definovaný jako řádná, samostatně definovaná veřejná třída v balíčku <code>b77_java_nz2._11_arrangement</code>	191
Výpis 11.2:	Prověření funkce třídy <code>Light</code> z výpisu 11.1.....	192
Výpis 11.3:	Definice třídy <code>ClassTemplate</code> v balíčku <code>b77_java_nz2._11_arrangement</code> představující šablonu definic třídy.....	200
Výpis 12.1:	Obsah souboru <code>package-info.java</code> v balíčku <code>b77_java_nz2._12_documentation</code>	206

Výpis 12.2:	Začátek definice třídy <code>Light</code> s doplněnými dokumentačními komentáři, ale bez komentářů oddělujících jednotlivé sekce	207
Výpis 13.1:	Možná definice metody <code>moveBy(int, int, IMovable)</code>	216
Výpis 13.2:	Definice interfejsu <code>IMovable</code>	217
Výpis 13.3:	Výňatek z definice třídy <code>Light</code> implementující interfejs <code>IMoveable</code>	219
Výpis 13.4:	Test přesovatelnosti instance třídy <code>Light</code>	220
Výpis 14.1:	Prověření plynulých změn pozice a velikosti instance třídy <code>Light</code>	224
Výpis 14.2:	Definice interfejsu <code>IChangeable</code> v balíčku <code>shapes77.canvas_4</code>	229
Výpis 14.3:	Definice interfejsu <code>IVehicle1_1</code> v balíčku <code>b77_java_nz2.vehicle</code>	231
Výpis 14.4:	Definice třídy <code>Robot1_1</code> implementující interfejs <code>IVehicle1_1</code> (po odebrání komentářů)	232
Výpis 14.5:	Test funkcionality třídy <code>Robot1_1</code>	234
Výpis 15.1:	Definice skriptu <code>importlib.jsh</code>	248
Výpis 15.2:	Definice interfejsu <code>IVehicle1_2</code> (po odebrání komentářů)	249
Výpis 15.3:	Provedené změny ve třídě <code>Robot1_2</code> (po odebrání komentářů)	249
Výpis 15.4:	Test upravené třídy <code>Robot1_2</code>	250
Výpis 16.1:	Ověření funkce příkazu <code>import static</code>	253
Výpis 16.2:	Definice interfejsu <code>shapes77.canvasmanager.ICMPaintable</code>	254
Výpis 16.3:	Definice záznamu a práce s ním	258
Výpis 16.4:	Definice třídy (záznamu) <code>Area</code>	259
Výpis 16.5:	Použití přepravek (záznamů) typu <code>Position</code> , <code>Size</code> a <code>Area</code>	260
Výpis 16.6:	Definice abstraktní továrny – interfejsu <code>IVehicleFactory1_3</code>	264
Výpis 16.7:	Definice instančních konstant a použití inicializačních bloků	266
Výpis 16.8:	Demonstrace vlivu použití inicializačních bloků	267
Výpis 16.9:	Definice statických konstant a inicializačních bloků	269
Výpis 16.10:	Aktivace třídy z výpisu 16.9	270
Výpis 16.11:	Třída <code>SwitchConst</code> používající přepínač	271
Výpis 16.12:	Spuštění třídy <code>SwitchConst</code> používající přepínač v podobě z výpisu 16.11	272
Výpis 16.13:	Třída <code>StaticArray</code> definující pole pevné velikosti jako statické atributy	272
Výpis 17.1:	Výpis přidaných částí kódu ve třídě <code>Light</code> v balíčku <code>b77_java_nz2._17_lambda</code>	277
Výpis 17.2:	Prověrka nezávislosti blikání instancí třídy <code>Light</code>	278
Výpis 17.3:	Lambda-výraz definovaný prostřednictvím operátoru <code>::</code>	280
Výpis 17.4:	Definice třídy <code>Interval</code>	283
Výpis 17.5:	Test funkcionality třídy <code>Interval</code>	283
Výpis 17.6:	Názvy a abstraktní metody vybraných funkčních interfejsů	287
Výpis 17.7:	Definice a použití lambda-výrazů za použití operátoru <code>::</code>	288
Výpis 17.8:	Přetypování lambda-výrazů	289
Výpis 18.1:	Definice prázdné třídy a přehled jejích členů	296
Výpis 18.2:	Počáteční primitivní definice třídy <code>Square</code>	298
Výpis 18.3:	Definice třídy <code>Square</code> s doplněným konstruktorem	300
Výpis 18.4:	Demonstrace přebíjení metod	302
Výpis 18.5:	Definice tříd <code>Matka</code> , <code>Dcera</code> , <code>Vnučka</code>	304
Výpis 18.6:	Definice metody <code>setSize(int, int)</code> pro třídu <code>Square</code>	305
Výpis 18.7:	Počátek chybové zprávy po žádosti o nastavení velikosti	306
Výpis 18.8:	Definice metody <code>setSize(int)</code> v interfejsu <code>IResizable</code>	306
Výpis 18.9:	Definice metody <code>setSize(int, int)</code> pro třídu <code>Square</code>	307

Výpis 18.10:	Demonstrace zakrývání statických metod.....	309
Výpis 19.1:	Experimenty s abstraktní třídou	315
Výpis 19.2:	Definice (hlavní) třídy Robot4_4 definující otočná vozidla s odebranými komentáři, přetíženými konstruktory a příkazy package a import	319
Výpis 19.3:	Definice abstraktního rodiče jednosměrných podobjektů – třídy ARobot1_4.....	323
Výpis 19.4:	Definice třídy Robot1E_4 definující jednosměrné podobjektoty otočené na východ.....	325
Výpis 19.5:	Příkazy a úryvky pro test otočných vozidel	327
Výpis 20.1:	Demonstrace chování instancí neměnného typu na příkladu stringů	334
Výpis 20.2:	Definice nešikovné verze metody reverseInWords (int)	338
Výpis 20.3:	Definice upravené verze metody reverseInWords (Long).....	339
Výpis 20.4:	Definice jednoduchého výčtového typu a jeho použití	340
Výpis 20.5:	Definice výčtového typu Direction robota Karla bez komentářů	341
Výpis 20.6:	Interval využívající komparátor	347
Výpis 21.1:	Ukázka vyhození výjimky a reakce systému na něj	350
Výpis 21.2:	Začátek definice třídy ExceptionThrowing	352
Výpis 21.3:	Reakce systému na vyhození výjimky ve třídě se zdrojovým kódem	352
Výpis 21.4:	Definice metody unreachable () demonstrující nedosažitelnost kódu za vyhozením výjimky	353
Výpis 21.5:	Metoda try_catch (int) demonstrující zachycení vyhozené výjimky.....	355
Výpis 21.6:	Demonstrace funkce bloku try...catch...finally	357
Výpis 21.7:	Definice vlastní kontrolované a nekontrolované výjimky.....	358
Výpis 21.8:	Demonstrace odezvy překladače při kontrole zabezpečení reakce na vyhození výjimky	360
Výpis 21.9:	Definice metody conversion () převádějící kontrolovanou výjimku na nekontrolovanou.....	361
Výpis 22.1:	Práce s množinou	369
Výpis 22.2:	Seznamy a práce s nimi	372
Výpis 22.3:	Dva způsoby seřazení prvků v seznamu	374
Výpis 22.4:	Základní operace s mapami.....	375
Výpis 23.1:	Demonstrace základních vlastností polí objektů	379
Výpis 23.2:	Základní vlastnosti polí hodnot primitivních typů	380
Výpis 23.3:	Inicializace pole a jeho tisk.....	382
Výpis 23.4:	Další způsoby inicializace	383
Výpis 23.5:	Dvozměrné pole	386
Výpis 23.6:	Definice a použití metody average (double . . .) využívající proměnný počet argumentů	387
Výpis 23.7:	Převod čísla do tisíce na vyjádření slovy v češtině	389
Výpis 24.1:	Srovnání použití lambda-výrazu a anonymní třídy.....	397
Výpis 24.2:	Použití metody forEach (Consumer<T>)	399
Výpis 24.3:	Omezení dvojtečkového cyklu for (:)	401
Výpis 24.4:	Třída RandomStrings – generátor pseudonáhodných textů	402
Výpis 24.5:	Test použitelnosti instancí RandomStrings v cyklu	403
Výpis 24.6:	Test použitelnosti instancí RandomStrings s interním iterátorem forEach ().....	404
Výpis 25.1:	Ukázka použití datovodů při generování čísel sportky.....	413
Výpis 26.1:	Několik příkladů práce s instancemi třídy File	421
Výpis 26.2:	Demonstrace použití výstupních datových proudů	428
Výpis 26.3:	Demonstrace použití vstupních datových proudů	431
Výpis 26.4:	Demonstrace použití třídy Scanner	432
Výpis 27.1:	Definice třídy GuessC představující jednoduchou aplikaci.....	435

Výpis 27.2:	Překlad a spuštění aplikace definované v jednom zdrojovém souboru	437
Výpis 27.3:	Definice třídy Main spouštějící demonstrační aplikaci	439
Výpis 27.4:	Definice třídy GuessIO umožňující vybrat způsob komunikace	440
Výpis 27.5:	Obsah souboru MANIFEST.MF ve složce 77_LIB.META-INF	442
Výpis 30.1:	Definice alternativního konstrukturu třídy ScenarioStep	476
Výpis 31.1:	Třída Portal v balíčku game77.ck1a_happy	483
Výpis 31.2:	Etapy vývoje aplikace a jejich identifikátory	484
Výpis 31.3:	Definice třídy ScenarioManager definující (mimo jiné) šťastný scénář	485
Výpis 31.4:	Zpráva obdržena po spuštění třídy game77.ck1a_happy.Portal	489
Výpis 31.5:	Definice počátku třídy game77.ck1b_duplet.ScenarioManager	491
Výpis 31.6:	Výsek zprávy o testu třídy game77.ck1b_duplet.Portal	492
Výpis 32.1:	Konstruktor instancí třídy Action	496
Výpis 32.2:	Začátek definice třídy AItemContainer s výchozí podobou konstrukturu a metody items()	498
Výpis 32.3:	Prozatímní definice třídy Bag	499
Výpis 32.4:	Definice konstrukturu instancí třídy Place	499
Výpis 32.5:	Prozatímní definice třídy World	500
Výpis 32.6:	Prozatímní definice třídy game77.ck1c_architecture.Game	502
Výpis 32.7:	Konec zprávy o testu třídy game77.ck1c_architecture.Portal	503
Výpis 33.1:	Definice upravených metod ve třídě Game	506
Výpis 33.2:	Definice statických metod isActive() a stop() ve třídě Action	507
Výpis 33.3:	Definice statické metody executeCommand(String) ve třídě Action	508
Výpis 33.4:	Definice statických metod executeEmptyCommand(), executeStandardCommand(String) a initialize() ve třídě Action	509
Výpis 33.5:	Úprava počátku definice třídy ScenarioManager v balíčku ck1d_start	511
Výpis 33.6:	Výsek zprávy po spuštění třídy ck1d_start.Portal	513
Výpis 34.1:	Upravený počátek definice třídy Place v balíčku game77.ck1e_world	515
Výpis 34.2:	Návrh definice metody pro inicializaci prostoru	516
Výpis 34.3:	Upravená verze třídy AItemContainer v balíčku game77.ck1e_world	516
Výpis 34.4:	Změny ve třídě ScenarioManager pro zpracování dalších konstant	518
Výpis 34.5:	Upravené metody ve třídě World v balíčku game77.ck1e_world	520
Výpis 34.6:	Upravená definice metody initialize() ve třídě game77.ck1e_world.Action	521
Výpis 34.7:	Výsek ze zprávy ze závěrečného testu na hladině Level.WORLD	521
Výpis 35.1:	Definice seznamu ACTIONS potřebných akcí	524
Výpis 35.2:	Nová definice funkce executeStandardCommand(String) v modulu actions	525
Výpis 35.3:	Definice metody move(String[]) realizující přesun do sousedního prostoru	526
Výpis 35.4:	Výsek klíčové části zprávy o provedení testu	526
Výpis 35.5:	Definice metody take(String[]) realizující přesun h-objektu z prostoru do batohu	527
Výpis 35.6:	Definice metody put(String[]) realizující přesun h-objektu z prostoru do batohu	527
Výpis 35.7:	Definice metody help(String[]) realizující nápovědu	528
Výpis 35.8:	Závěr zprávy o testu na hladině Level.BASIC	528
Výpis 36.1:	Typy kroků, které je třeba zařadit do chybového scénáře	532
Výpis 36.2:	Začátek definice chybového scénáře	533
Výpis 36.3:	Závěr zprávy o testu na hladině TRIPLET	534

Výpis 36.4:	Upravená definice statické metody <code>executeCommand(String)</code> ve třídě <code>Action</code>	535
Výpis 36.5:	Upravená definice statické metody <code>executeStandardCommand(String)</code>	535
Výpis 36.6:	Upravená definice statické metody <code>move(String[])</code>	536
Výpis 36.7:	Upravená definice třídy <code>Item</code>	538
Výpis 36.8:	Upravená definice konstruktora instancí třídy <code>World</code>	538
Výpis 36.9:	Upravené definice metod <code>addItem(Item)</code> , <code>removeItem(Item)</code> a <code>initialize()</code> ve třídě <code>Bag</code>	540
Výpis 36.10:	Upravené definice metody <code>take(String[])</code> ve třídě <code>Action</code>	540
Výpis 36.11:	Upravené definice metody <code>put(String[])</code> ve třídě <code>Action</code>	541
Výpis 37.1:	Kroky scénáře <code>HAPPY</code> s nestandardními příkazy a definovanými konstantami	544
Výpis 37.2:	Finální definice seznamu <code>ACTIONS</code> všech instancí třídy <code>Action</code>	544
Výpis 37.3:	Výchozí podoba definic nestandardních akcí	545
Výpis 37.4:	Upravený startovní krok všech scénářů ve třídě <code>SceneManager</code>	548
Výpis 37.5:	Upravené nestandardní kroky ve scénáři <code>HAPPY</code>	549
Výpis 37.6:	Typy kroků se špatně zadanými příkazy, jež je třeba prověřit	549
Výpis 37.7:	Počátek scénáře <code>MISTAKES_NS</code>	551
Výpis 37.8:	Definice metod <code>conditions()</code> a <code>tests()</code> ve třídě <code>Game</code>	552
Výpis 37.9:	Definice atributů <code>CONDITIONS</code> a <code>TESTS</code> ve třídě <code>Action</code>	553
Výpis 37.10:	Upravená metoda <code>initialize()</code> ve třídě <code>Action</code>	553
Výpis 37.11:	Upravená definice metody <code>wakeUp(String[])</code> ve třídě <code>Action</code>	554
Výpis 38.1:	Výchozí definice třídy <code>Main</code> v balíčku <code>game77.ckl_user_io</code>	556
Výpis 38.2:	Definice metody <code>multirun(IGame)</code>	557
Výpis 38.3:	Definice funkce <code>currentState(IGame)</code>	558
Výpis 38.4:	Definice interfejsu <code>IUI</code>	560
Výpis 38.5:	Upravená definice metody <code>main(String[])</code> ve třídě <code>Main2</code>	561
Výpis 38.6:	Upravené definice metod <code>run(...)</code> a <code>multirun(...)</code> ve třídě <code>Main2</code>	561
Výpis 38.7:	Definice třídy <code>Console</code> v modulu <code>interfaces</code>	562
Výpis 38.8:	Definice třídy <code>PrimitiveGUI</code>	565

Seznam S2

Seznam obrázků

Obrázek 1.1: Průběh popularity nejpopulárnějších IDE (zdroj: https://www.jrebel.com/blog/best-java-ide)	43
Obrázek 2.1: Konzolové okno otevřené po spuštění <code>dávky !_JShell.bat</code> na mém počítači	46
Obrázek 5.1: Vytvořený svět spolu s robotem.....	100
Obrázek 6.1: Stromová struktura balíčků	104
Obrázek 6.2: Svět Robota po provedení akcí z výpisu 6.5	113
Obrázek 6.3: Svět Robota po provedení akcí z výpisu 6.6	113
Obrázek 7.1: Výsledná podoba dvorku se čtyřmi roboty.....	133
Obrázek 8.1: Vývojový diagram cyklu s počáteční podmínkou (cyklu <code>while</code>).....	141
Obrázek 8.2: Syntaktický diagram cyklu s počáteční podmínkou (cyklu <code>while</code>).....	142
Obrázek 8.3: Vývojový diagram cyklu s ukončovací podmínkou (cyklu <code>do...while</code>).....	143
Obrázek 8.4: Syntaktický diagram cyklu s ukončovací podmínkou (cyklu <code>do...while</code>)	144
Obrázek 8.5: Vývojový diagram cyklu s parametrem (cyklu <code>for</code>)	145
Obrázek 8.6: Syntaktický diagram cyklu s parametrem (cyklu <code>for</code>)	146
Obrázek 8.7: Vývojový diagram cyklu s podmínkou uprostřed	150
Obrázek 9.1: Vývojový diagram podmíněného příkazu.....	153
Obrázek 9.2: Vývojový diagram úplného podmíněného příkazu.....	154
Obrázek 9.3: Syntaktický diagram obecného podmíněného příkazu	154
Obrázek 9.4: Vývojový diagram složeného podmíněného příkazu	156
Obrázek 9.5: Bludiště s robotem	156
Obrázek 9.6: Vývojový diagram přepínače – příkazu <code>switch</code>	159
Obrázek 9.7: Syntaktický diagram přepínače – příkazu <code>switch</code>	159
Obrázek 10.1: Diagram tříd balíčku <code>shapes77.canvas_1</code>	176
Obrázek 10.2: Okno plátna se třemi vytvořenými světly	177
Obrázek 13.1: Diagram tříd balíčku <code>shapes77.canvas_2</code>	215
Obrázek 14.1: Diagram tříd balíčku <code>shapes77.canvas_3</code>	223
Obrázek 14.2: Diagram tříd balíčku <code>shapes77.canvas_4</code>	227
Obrázek 14.3: Diagram tříd balíčku <code>shapes77.canvas_4</code> po odstranění šipek závislostí a přesunu ikon tříd a interfejsů do přehlednějšího uspořádání	228
Obrázek 15.1: Zmenšení počtu závislostí po aplikaci vzoru Prostředník	243
Obrázek 15.2: Podoba diagramu tříd po zavedení abstrakce, které se musejí všechny komunikující objekty přizpůsobit	245
Obrázek 15.3: Diagram tříd balíčku <code>shapes77.canvasmanager</code>	248
Obrázek 16.1: Diagramy tříd možných implementací vzoru Adaptér	256

Obrázek 17.1	285
Obrázek 18.1	Rodičovský podobjekt 298
Obrázek 18.2:	Schematická demonstrace vnitřního uspořádání instance třídy Vnučka 304
Obrázek 18.3:	Malý popisek 305
Obrázek 19.1:	Diagram tříd balíčku shapes77.geom 317
Obrázek 19.2:	Principiální diagram uspořádání tříd podle návrhového vzoru Stav 318
Obrázek 19.3:	Diagram tříd balíčku b77_java_nz2._19_abstract zobrazující třídu Robot4_4 a její pomocné třídy 326
Obrázek 21.1:	Hierarchie dědění výjimek 354
Obrázek 22.1:	Hlavní datové typy patřící do knihovny kolekcí 364
Obrázek 22.2:	Syntaktický diagram „dvojtečkového“ cyklu for (cyklu „for each“) 368
Obrázek 24.1:	Rozdělení interních datových typů 392
Obrázek 25.1:	Schéma postupu zpracování dat předávaných datovody 410
Obrázek 26.1:	Exponenciální nárůst počtu tříd při přidávání funkčnosti 423
Obrázek 26.2:	Třídy po aplikaci dekorátorů 424
Obrázek 29.1:	Předběžný diagram tříd balíčku game77.api 467
Obrázek 30.1:	Předběžný diagram tříd balíčku game77.api 480
Obrázek 38.1:	Dialogové okno s výzvou k zadání údajů 566
Obrázek 38.2:	Diagram tříd aktuálního stavu balíčku game77.ck1l_user_io 567

Seznam S3

Seznam tabulek

Tabulka 4.1: Přehled operátorů, jejich priorit a asociativity	79
Tabulka 4.2: Inkrementační a dekrementační operátory	87
Tabulka 20.1: Primitivní a obalové typy	335
Tabulka 26.1: Rodičovské třídy jednotlivých typů datových proudů	425
Tabulka 28.1: Porovnání metody shora dolů a zdola nahoru	453

Seznam S4

Seznam odboček – podšeděných bloků

Odbočka – podšeděný blok	29
Datová struktura strom	105
Historie robota Karla	111
Tisk na standardní výstup	126
Zásobník návratových adres – ZNA.....	128
Bílé znaky a uspořádání programu	173
Problémy s kódováním znaků	337
Prohlížení obsahu JAR-souborů	441
Co to je h-objekt.....	458
Návrhový vzor Fasáda.....	564