

Část G

Seznamy

Tato část obsahuje seznamy hypertextových odkazů na výpisy programů, obrázky, tabulky a odbočky – podšeděné bloky. Není součástí tištěné knihy, aby zbytečně nezvyšovala její cenu. Je však součástí elektronických verzí, v nichž můžete hypertextové odkazy efektivně využít.

Seznam výpisů programů

Výpis 1.1:	Komunikace s interpretem spuštěným v konzolovém okně	39
Výpis 1.2:	Reakce na odsazení příkazu	43
Výpis 2.1:	Reakce na zadání velkého čísla	52
Výpis 2.2:	Zpřehlednění čísel vložením podtržitek	52
Výpis 2.3:	Zadávání a zobrazování desetinných čísel	53
Výpis 2.4:	Zadávání a zobrazování komplexních čísel	53
Výpis 2.5:	Zápis čísel v alternativních číselných soustavách	54
Výpis 2.6:	Reakce na použití objektů None a ... (Ellipsis)	56
Výpis 3.1:	Zápis jednořádkového textu	58
Výpis 3.2:	Zápis víceřádkového textu	59
Výpis 3.3:	Zadání znaků pomocí escape sekvencí	61
Výpis 3.4:	Komentáře	63
Výpis 3.5:	Zadání na více řádcích	63
Výpis 4.1:	Zadání argumentu nápovědy pomocí stringu	78
Výpis 4.2:	Přepnutí do režimu nápovědy a zpět	79
Výpis 4.3:	Volání funkce rozepsané na více řádcích	80
Výpis 5.1:	Ukázky tří druhů dělení	86
Výpis 5.2:	Komplexní číslo jako exponent – Eulerova rovnost	87
Výpis 5.3:	Nekonečna a nesmyslná čísla	88
Výpis 5.4:	Sčítání textů	89
Výpis 5.5:	Násobení textů	89
Výpis 5.6:	Indexace stringů	90
Výpis 6.1:	Přiřazení hodnoty proměnné	96
Výpis 6.2:	Skrytá běhová chyba	97
Výpis 6.3:	Současné přiřazení skupiny hodnot proměnným	98
Výpis 6.4:	Zavedení proměnných <code>in</code> a <code>nan</code>	99
Výpis 6.5:	Ukázky vnořeného volání funkcí	100
Výpis 6.6:	Zjištění typu objektu	101
Výpis 6.7:	Přiřazení funkcí (přesněji odkazů na funkce) do proměnných	102
Výpis 6.8:	Definice a použití lambda-výrazu	103
Výpis 6.9:	Přirazovací výraz a jeho použití	106
Výpis 6.10:	Ovlivnění toho, bude-li se zobrazovat systémový či uživatelský podpis	108
Výpis 7.1:	Převody logických hodnot na čísla a jiných hodnot na logické	110
Výpis 7.2:	Zřetěžené porovnávání	112
Výpis 7.3:	Test totožnosti objektů	113
Výpis 7.4:	Chování logických operátorů	114
Výpis 7.5:	Vliv priorit na výsledek	115
Výpis 7.6:	Chování bitových operátorů	116
Výpis 7.7:	Bitové posuny	117
Výpis 7.8:	Použití podmíněného výrazu	118

Výpis 8.1:	Složené přiřazovací operátory	122
Výpis 8.2:	Použití příkazu <code>del</code>	123
Výpis 8.3:	Použití příkazu <code>assert</code>	125
Výpis 8.4:	Spuštění systému Python s aktivovaným a deaktivovaným příkazem <code>assert</code>	126
Výpis 8.5:	Kontrola shodnosti odsazení příkazů	126
Výpis 9.1:	Importování modulů	134
Výpis 9.2:	Přejmenování importovaného modulu	135
Výpis 9.3:	Import zadaných objektů ze zadaného modulu	137
Výpis 9.4:	Použitelnost proměnné s odkazem na modul	138
Výpis 9.5:	Import všech objektů ze zadaného modulu	139
Výpis 9.6:	Demonstrace nárůstu počtu atributů po hromadném importu	139
Výpis 9.7:	Modul <code>builtins</code>	142
Výpis 10.1:	Modul <code>m10a_Modul</code> sloužící k demonstraci chování Pythonu při zavádění modulu	145
Výpis 10.2:	Import modulu <code>m10a_Modul</code>	150
Výpis 10.3:	Nové načtení modulu <code>m10a_Modul</code>	154
Výpis 10.4:	„Neaktualizace“ přímo importovaných odkazů	154
Výpis 10.5:	Opětne načtení verze s chybou	155
Výpis 10.6:	Definice modulu <code>m10b_Demo_all</code> vyjmenovávajícího veřejné atributy	157
Výpis 10.7:	Hvězdičkový import modulu <code>m10b_Demo_all</code>	158
Výpis 10.8:	Import modulu <code>posixpath</code> nebo <code>npath</code> zprostředkovaný modulem <code>os</code>	159
Výpis 10.9:	Prověrka zprostředkování importu	159
Výpis 10.10:	Modul <code>m10c_Circular_C</code> importující modul <code>m10d_Circular_D</code>	160
Výpis 10.11:	Modul <code>m10d_Circular_D</code> importující modul <code>m10c_Circular_C</code>	160
Výpis 10.12:	Pokus o import modulu <code>m10c_Circular_C</code> a reakce na něj	161
Výpis 11.1:	Jednořádková definice funkce	164
Výpis 11.2:	Víceřádková definice funkce	165
Výpis 11.3:	Prisazení definice funkce v interaktivním režimu	168
Výpis 11.4:	Modul <code>m11a_Funkce_v_modulu</code> demonstrující pravidla pro definice funkcí v modulu	169
Výpis 11.5:	Dynamická povaha jazyka při definici a volání funkcí	170
Výpis 11.6:	Definice modulu <code>m11b_dbg_demo</code>	172
Výpis 11.7:	Import modulu <code>m11b_dbg_demo</code> a zavolání jeho funkce <code>fce1()</code>	173
Výpis 12.1:	Definice a volání funkce s parametrem	175
Výpis 12.2:	Definice funkce s lokálními proměnnými	176
Výpis 12.3:	Volání funkcí s parametry	176
Výpis 12.4:	Funkce s povinně pojmenovanými argumenty	177
Výpis 12.5:	Funkce s povinně pozičními argumenty	178
Výpis 12.6:	Funkce s různými kombinacemi povinných parametrů	180
Výpis 12.7:	Volání funkce <code>print()</code> s vlastní hodnotou implicitního argumentu	181
Výpis 12.8:	Definice funkce se dvěma implicitními argumenty a možností jejího volání	182
Výpis 12.9:	Konstantnost předdefinovaných hodnot	182
Výpis 12.10:	Funkce s vedlejším efektem	183
Výpis 12.11:	Definice a použití funkcí vracejících hodnotu	184
Výpis 12.12:	Nepoužitelnost přetěžování funkcí	184
Výpis 12.13:	Deklarace typů proměnných a návratových hodnot prostřednictvím anotací	186
Výpis 13.1:	Definice vnitřních funkcí	188
Výpis 13.2:	Zakrytí globální proměnné stejnojmennou lokální	192
Výpis 13.3:	Použití příkazu <code>global</code>	193
Výpis 13.4:	Použití příkazu <code>non local</code>	195

Výpis 13.5:	Demonstrace vnořeného volání funkcí.....	197
Výpis 13.6:	Použití funkce vracející funkci.....	198
Výpis 13.7:	Využití atributů funkce.....	199
Výpis 13.8:	Nelokální proměnné a uzávěry.....	200
Výpis 13.9:	Definice modulu <code>m13b_Circular_B</code> importujícího modul <code>m13c_Circular_C</code>	201
Výpis 13.10:	Definice modulu <code>m13c_Circular_C</code> importujícího modul <code>m13b_Circular_B</code>	201
Výpis 13.11:	Import modulu <code>m13b_Circular_B</code>	202
Výpis 14.1:	Ukázka použití jednoduchého podmíněného příkazu.....	204
Výpis 14.2:	Ukázka použití úplného podmíněného příkazu.....	205
Výpis 14.3:	Ukázky použití rozšířeného podmíněného příkazu.....	207
Výpis 14.4:	Řešení jednoduché úlohy prostřednictvím přepínače.....	208
Výpis 14.5:	Ukázka možného sdružování hodnot ve vzorech.....	209
Výpis 14.6:	Přímé zadání podmíněného příkazu v interaktivním režimu.....	210
Výpis 15.1:	Ukázky definic funkcí používajících rekurzivní volání.....	212
Výpis 15.2:	Rekurzivní definice faktoriálu.....	213
Výpis 15.3:	Ukázky definic funkcí používajících cyklus <code>while</code>	214
Výpis 15.4:	Nekonečný cyklus.....	215
Výpis 15.5:	Použití příkazu <code>break</code>	216
Výpis 15.6:	Použití přiřazovacího výrazu v hlavičce cyklu <code>while</code>	218
Výpis 15.7:	Cyklus <code>while</code> využívající větve <code>else</code>	219
Výpis 15.8:	Použití příkazu <code>continue</code>	220
Výpis 15.9:	Použití příkazu <code>for</code> se zdrojem definovaným jako sada hodnot.....	221
Výpis 15.10:	Použití příkazu <code>for</code> se zdrojem generovaným funkcí <code>range()</code>	222
Výpis 15.11:	Indexace znaků v textovém řetězci – stringu.....	223
Výpis 15.12:	Zpracování jednotlivých znaků v textovém řetězci bez použití indexů.....	223
Výpis 15.13:	Ukázka zanořování cyklů.....	224
Výpis 15.14:	Příkaz <code>for</code> s postupným použitím více zdrojů.....	225
Výpis 15.15:	Příkaz <code>for</code> s větví <code>else</code>	225
Výpis 16.1:	Ukázky syntaktických chyb a reakcí na ně při zadávání příkazů v prostředí IDLE.....	228
Výpis 16.2:	Ukázky syntaktických chyb a reakcí na ně při zadávání příkazů z konzolového okna.....	230
Výpis 16.3:	Zopakovaný výpis 10.5 ze strany 157 zobrazující načtení třetí verze modulu <code>m10a_Modul</code> z výpisu 10.1 na straně 147.....	231
Výpis 16.4:	Ukázka zachycení a ošetření výjimky.....	234
Výpis 16.5:	AHA-příklad používající příkaz <code>try</code> se všemi dostupnými větvemi.....	238
Výpis 16.6:	Měření rychlosti počítače a přesnosti odhadu času.....	239
Výpis 16.7:	Zdánlivé záludnosti větve <code>finally</code>	240
Výpis 16.8:	Funkce <code>assert_stmt()</code> demonstrující funkci příkazu <code>assert</code>	242
Výpis 17.1:	Demonstrace neměnnosti objektů.....	245
Výpis 17.2:	Vytváření seznamů pomocí literálů.....	247
Výpis 17.3:	Vytváření seznamů pomocí konstruktora <code>list()</code>	248
Výpis 17.4:	Vytváření seznamů z jiných seznamů pomocí sčítání a násobení.....	249
Výpis 17.5:	Vytvoření seznamu pomocí generátorové notace.....	250
Výpis 17.6:	Použití funkcí <code>append()</code> a <code>extend()</code>	252
Výpis 17.7:	Demonstrace chování proměnných objektů.....	253
Výpis 17.8:	Postupné budování seznamu.....	254
Výpis 17.9:	Rozšiřování seznamu přičítáním.....	254
Výpis 17.10:	Chybné pokusy o rozšíření seznamu.....	255
Výpis 17.11:	Ukázky použití modifikačních funkcí a metod pracujících s indexy.....	256

Výpis 17.12:	Ukázky použití modifikačních metod pracujících s celým seznamem.....	258
Výpis 17.13:	Modul m17a_Pascal s funkcemi pracujícími s vícerozměrnými seznamy	259
Výpis 17.14:	Načtení modulu m17a_Pascal a prověření jeho funkce	259
Výpis 17.15:	Anotace odkazující na seznamy	260
Výpis 18.1:	Vytváření n-tic pomocí literálů	262
Výpis 18.2:	Vytváření n-tic pomocí konstrukturu tuple ()	264
Výpis 18.3:	Vytváření n-tic z jiných n-tic pomocí sčítání a násobení	264
Výpis 18.4:	Vytváření n-tic přičítáním	265
Výpis 18.5:	Balení a rozbalování n-tic a seznamů	266
Výpis 18.6:	Prohazování hodnot proměnných	267
Výpis 18.7:	Hvězdičkové pravidlo	267
Výpis 18.8:	Vytvoření n-tice pomocí generátoru	268
Výpis 18.9:	Přístup k prvkům n-tic	270
Výpis 18.10:	Sčítání seznamů a n-tic	270
Výpis 18.11:	Chování n-tic s proměnnými a neměnnými prvky	271
Výpis 18.12:	Prvky standardních n-tic nemají jména	272
Výpis 18.13:	Pojmenované n-tice	272
Výpis 18.14:	Pojmenované n-tice s implicitními argumenty	273
Výpis 19.1:	Vytváření množin pomocí literálů	276
Výpis 19.2:	Vytváření množin pomocí konstrukturu set ()	277
Výpis 19.3:	Povolené a nepovolené argumenty konstrukturu set ()	277
Výpis 19.4:	Vytváření množin prostřednictvím množinových operací	278
Výpis 19.5:	Vytváření množin pomocí generátorové notace	280
Výpis 19.6:	Vytváření zmrazených množin	281
Výpis 19.7:	Jednoprvkové modifikační množinové operace	282
Výpis 19.8:	Modifikace množin množinami či jinými zdroji	283
Výpis 19.9:	Porovnání množin	285
Výpis 20.1:	Vytváření slovníků pomocí literálů	287
Výpis 20.2:	Vytváření slovníků pomocí konstrukturu dict ()	288
Výpis 20.3:	Vytváření slovníků pomocí funkce fromkeys ()	290
Výpis 20.4:	Vytváření slovníků pomocí generátorové notace	290
Výpis 20.5:	Přístup k položkám prostřednictvím klíčů	292
Výpis 20.6:	Metody pro práci s jednotlivými položkami	293
Výpis 20.7:	Použití metody update ()	294
Výpis 20.8:	Pohledy	295
Výpis 20.9:	Pohledy jako generátory	296
Výpis 20.10:	Operace s pohledy	297
Výpis 21.1:	Demonstrace chování programu při předávání argumentů hodnotou	299
Výpis 21.2:	Definice a ukázky použití funkce gr ()	300
Výpis 21.3:	Zpracování argumentů při použití hvězdičkového parametru	302
Výpis 21.4:	Použití a zpracování hvězdičkového argumentu	303
Výpis 21.5:	Zpracování argumentů při použití dvouhvězdičkového parametru	304
Výpis 21.6:	Zpracování dvouhvězdičkového argumentu	305
Výpis 21.7:	Příklady zpracování argumentů inspirované [5]	306
Výpis 21.8:	Možné použití hvězdičkových a dvojhvězdičkových argumentů	307
Výpis 22.1:	Použití formátovacího operátoru %	313
Výpis 22.2:	Použití prázdné formátové položky	315
Výpis 22.3:	Různé způsoby zadávání nahrazovaného textu	317
Výpis 22.4:	Efekt zadání konverze konverzí	318
Výpis 22.5:	Zadání požadovaného minimálního počtu pozic	319

Výpis 22.6:	Zadání požadované přesnosti, resp. maximálního počtu pozic	320
Výpis 22.7:	Typy formátů pro celočíselné hodnoty	321
Výpis 22.8:	Typy formátů pro zobrazení reálných čísel	322
Výpis 22.9:	Použití oddělovače skupin číslic	323
Výpis 22.10:	Zadání oddělovačů skupin číslic a alternativního formátu	324
Výpis 22.11:	Zarovnání a plnění	326
Výpis 22.12:	Používání vnořených nahrazovacích polí	327
Výpis 22.13:	Definice funkce <code>printpasf()</code> v modulu <code>m22a_PrintPasF</code>	328
Výpis 22.14:	Použití funkce <code>printpasf()</code> v modulu <code>m22a_PrintPasF</code>	329
Výpis 22.15:	Trasovací funkce <code>prSE()</code> v modulu <code>dbg</code>	330
Výpis 22.16:	Trasovací funkce <code>prIN()</code> v modulu <code>dbg</code>	331
Výpis 22.17:	Ukázka použití trasovacích funkcí <code>prSE()</code> a <code>prIN()</code>	332
Výpis 23.1:	Hluboké a mělké kopie	335
Výpis 23.2:	Použití operátorů <code>in</code> a <code>not in</code>	338
Výpis 23.3:	Použití funkce <code>reversed()</code>	339
Výpis 23.4:	Použití funkce <code>sorted()</code>	339
Výpis 23.5:	Indexování a vykrajování u rozsahů – objektů typu <code>range</code>	341
Výpis 23.6:	Modifikace obsahu posloupností	342
Výpis 23.7:	Postup při výměně prvku v <code>n</code> -tici	343
Výpis 24.1:	Zjištění a změna aktuální složky	349
Výpis 24.2:	Syntéza a analýza cest	351
Výpis 24.3:	Vytváření a mazání složek	352
Výpis 24.4:	Získání informací o souborech	353
Výpis 24.5:	Otevření souboru	355
Výpis 24.6:	Zápis dat, splachování a zavírání souborů	356
Výpis 24.7:	Čtení souborů	359
Výpis 26.1:	Definice třídy a jejích atributů	375
Výpis 26.2:	Práce s atributy objektu	377
Výpis 26.3:	Přidávání a odstraňování atributů objektu	377
Výpis 26.4:	Přidávání a odstraňování metody	378
Výpis 26.5:	Třída jako parametr funkce	379
Výpis 26.6:	Vytvoření prvních instancí	380
Výpis 26.7:	Kvalifikace atributu třídy instancí	381
Výpis 26.8:	Možná použití dekorátoru	383
Výpis 26.9:	Přidání a použití instanční metody	384
Výpis 26.10:	Definice třídy <code>C2</code>	385
Výpis 26.11:	Vytváření instancí třídy <code>C2</code> a používání jejich atributů	387
Výpis 26.12:	Vytváření nových instančních datových atributů	388
Výpis 26.13:	Změny třídních datových atributů	389
Výpis 26.14:	Třídni datový atribut jako implicitní hodnota instančního	390
Výpis 26.15:	Zavádění nových třídních funkčních atributů	391
Výpis 26.16:	Zavádění nových instančních funkčních atributů	391
Výpis 26.17:	Demonstrace nutnosti kvalifikace atributů	393
Výpis 26.18:	Ukázky použití systémových atributů	395
Výpis 27.1:	Definice tříd <code>LA</code> , <code>LB</code> a <code>LC</code> v modulu <code>m27a_LABC</code>	399
Výpis 27.2:	Jmenné prostory a dosažitelné atributy tříd z modulu <code>m27a_LABC</code>	401
Výpis 27.3:	Jmenné prostory a dosažitelné atributy instancí tříd z modulu <code>m27a_LABC</code>	402
Výpis 27.4:	Volání metod v hierarchii dědění	403
Výpis 27.5:	Demonstrace dědění initorů	404

Výpis 27.6:	Ukázka definice a použití vlastní výjimky	406
Výpis 28.1:	Zdrojový kód tříd z modulu <code>m28a_Diament</code> demonstrujících chování tříd uspořádaných do diamantové architektury	409
Výpis 28.2:	Prověrka vlastností násobného dědění	410
Výpis 28.3:	Definice třídy <code>Dcera2</code> v modulu <code>m28a_Diament</code>	412
Výpis 28.4:	Použití třídy <code>Dcera2</code>	412
Výpis 28.5:	Definice modulu <code>m28b_Arguments</code> s třídami používajícími násobné dědění a konstruktory s parametry	413
Výpis 28.6:	Průběh inicializace instance třídy <code>DceraA</code>	414
Výpis 28.7:	Nerealizovatelná a následně opravená hierarchie tříd	416
Výpis 28.8:	Definice složitější hierarchie tříd	416
Výpis 28.9:	Pseudosoukromý atribut a komolení jmen	418
Výpis 28.10:	Definice tříd <code>Bc</code> a <code>Mc</code> v modulu <code>m28c_Komolení</code>	419
Výpis 28.11:	Použití tříd <code>Bc</code> a <code>Mc</code> a jejich instancí	420
Výpis 29.1:	Přímá definice vlastností jako instance třídy <code>property</code>	423
Výpis 29.2:	Definice vlastností pomocí <code>lambda</code> -výrazů	425
Výpis 29.3:	Důsledky změny třídního atributu definujícího vlastnost	426
Výpis 29.4:	Zadání vlastnosti pomocí dekorátoru	427
Výpis 29.5:	Demonstrace použití vlastností instancí třídy <code>C29c</code>	428
Výpis 29.6:	Definice formálně abstraktní třídy <code>Matka29a</code> bez abstraktních metod	430
Výpis 29.7:	Definice abstraktní třídy <code>Matka29b</code> s abstraktní metodou	431
Výpis 29.8:	Definice potomka abstraktní třídy nepřebíjejícího abstraktní metodu	432
Výpis 29.9:	Definice třídy <code>Dcera29b</code> , jež je potomkem abstraktní třídy <code>Matka29b</code>	432
Výpis 29.10:	Definice dalších druhů abstraktních metod	434
Výpis 30.1:	Vytvoření výčtového datového typu prostřednictvím volání funkce <code>Enum()</code> a demonstrace vlastností daného typu a jeho atributů–instancí	439
Výpis 30.2:	Definice výčtového datového typu jako třídy	441
Výpis 30.3:	Definice výčtového typu <code>Směr</code> demonstrujícího některé další možnosti	442
Výpis 30.4:	Automatické přiřazování obalovaných hodnot	443
Výpis 30.5:	Poloautomatické nastavování obalovaných hodnot	444
Výpis 30.6:	Definice datové třídy a její test	446
Výpis 30.7:	Získání hodnot v rámci vyhodnocování vzorů v příkazu <code>match</code>	448
Výpis 30.8:	Prověřování posloupností	449
Výpis 30.9:	Použití vzoru s výběrem hodnot	450
Výpis 31.1:	Šablona initorů standardních balíčků uložená v souboru <code>__init__.py</code> v kořenovém balíčku	453
Výpis 31.2:	Výkonný kód modulu <code>m1_cls</code> v balíčku <code>PKG.sub_pkg.subsub_pkg</code>	453
Výpis 31.3:	Import modulu <code>PKG.sub_pkg.subsub_pkg.m1_cls</code> a jeho opětné načtení po úpravě	454
Výpis 31.4:	Kód modulu <code>PKG.sub_pkg.subsub_pkg.m2_dot</code> bez úvodních a závěrečných tisků	455
Výpis 31.5:	Import modulu <code>PKG.sub_pkg.subsub_pkg.m2_dot</code>	456
Výpis 31.6:	Kód modulu <code>PKG.sub_pkg.subsub_pkg.m3_star</code> , tentokrát včetně úvodních a závěrečných tisků	457
Výpis 31.7:	Průběh <code>m3_star run()</code>	458
Výpis 31.8:	Obsah souboru <code>__init__.py</code> v balíčku <code>PKG.sub_pkg.subsub_pkg</code>	459
Výpis 31.9:	Průběh znovuzavedení aktualizovaného balíčku <code>PKG.sub_pkg.subsub_pkg</code>	459
Výpis 31.10:	Průběh znovuzavedení aktualizovaného balíčku <code>m3_star</code>	460

Výpis 31.11:	Definice modulu <code>mA_importing</code> v balíčku <code>NS.ns_pkg</code> ve standardním zdroji.....	463
Výpis 31.12:	Definice modulu <code>mB_imported</code> v balíčku <code>NS.ns_pkg</code> v externím zdroji.....	463
Výpis 31.13:	Zavedení modulu <code>mA_importing</code>	464
Výpis 31.14:	Definice souboru <code>__init__</code> – initoru balíčku <code>NS.pkg</code> ve stromu <code>68_NS</code>	466
Výpis 31.15:	Import atributu balíčku <code>NS.pkg</code>	466
Výpis 32.1:	Definice modulu <code>m32a_Script</code>	468
Výpis 32.2:	Reakce na spuštění modulu <code>m32a_Script</code> v interaktivním režimu.....	469
Výpis 32.3:	Reakce na spuštění modulu <code>m32a_Script</code> v příkazovém okně <code>Windows</code>	469
Výpis 32.4:	Modul <code>m4_run</code> v balíčku <code>PKG.sub_pkg.subsub_pkg</code>	470
Výpis 32.5:	Vytvoření samostatné aplikace pomocí modu <code>wi_napp</code> a její spuštění.....	471
Výpis 32.6:	Definice modulu <code>__main__</code> v souboru <code>PKG.pyz</code>	471
Výpis 32.7:	Definice funkce <code>main()</code> v modulu <code>m32b_APR</code>	473
Výpis 32.8:	Spouštění skriptu <code>m32b_APR</code>	474
Výpis 33.1:	Definice třídy <code>Prime</code> , její instance jsou iterovatelné objekty a současně iterátory.....	480
Výpis 33.2:	Generátorový výraz a jeho použití.....	481
Výpis 33.3:	Nekorektní použití generátorového výrazu.....	482
Výpis 33.4:	Definice a použití generátorové funkce <code>prime_fun()</code>	484
Výpis 33.5:	Definice generátorové funkce <code>gen1()</code> demonstrující vícenásobné použití příkazu <code>yield</code>	485
Výpis 33.6:	Demonstrace prostřednictvím ruční aktivace výrazu <code>yield</code>	488
Výpis 33.7:	Definice generátorové funkce <code>ord_gen()</code> pracující s výrazem <code>yield</code>	490
Výpis 33.8:	Definice funkce <code>orders()</code> využívající možnost ovlivnit generátor.....	491
Výpis 33.9:	AHA-příklad s definicí generátorových funkcí <code>gen2()</code> a <code>gen21()</code> demonstrujících použití výrazů <code>yield</code> a <code>yield from</code>	493
Výpis 34.1:	Využití dekorátoru <code>total_ordering</code> pro doplnění porovnávacích metod.....	497
Výpis 34.2:	Definice třídy <code>Zlomky</code> v modulu <code>m34a_Zlomky</code>	501
Výpis 34.3:	Práce s instancemi třídy <code>Zlomky</code>	502
Výpis 34.4:	Definice třídy <code>Indexer</code> přetěžující operátor <code>[]</code>	504
Výpis 34.5:	Přidání instanční metody pro konkrétní instanci.....	505
Výpis 34.6:	Demonstrace rozdílů v kódu při použití příkazu <code>with</code>	507
Výpis 34.7:	Emulace postupu při zadání příkazu <code>with</code>	508
Výpis 35.1:	Anotace parametrů funkce, její návratové hodnoty a atributů modulu.....	512
Výpis 35.2:	Zapamatované anotace při okamžitém vyhodnocení platné doposud.....	513
Výpis 35.3:	Zapamatované anotace při odloženém vyhodnocení plánované od verze 3.11.....	514
Výpis 35.4:	Co o anotacích prozradí nápověda.....	515
Výpis 35.5:	Definice a použití přezdivek datových typů.....	517
Výpis 35.6:	Slučování datových typů.....	518
Výpis 36.1:	Ukázka použití třídnicích metod.....	522
Výpis 36.2:	Ukázka definice tovární metody.....	524
Výpis 36.3:	Definice dekorátoru – třídy <code>traced</code> v modulu <code>m36a_Traced</code>	525
Výpis 36.4:	Použití dekorátoru <code>traced</code>	526
Výpis 36.5:	Definice dekorátorů <code>jedináčka</code> v modulu <code>m36c_Singleton</code>	529
Výpis 36.6:	Použití dekorátoru <code>singleton</code>	530
Výpis 37.1:	Ekvivalent definice vestavěné třídy <code>property</code> v modulu <code>m37a_Property</code>	533
Výpis 37.2:	Definice třídy <code>descriptor</code>	534
Výpis 37.3:	Hrátky s deskriptorem.....	536
Výpis 37.4:	Postup metody <code>__getattr__()</code> při vyhodnocování atributu instance.....	538

Výpis 37.5:	Deskriptor pro odložené vyhodnocení.....	539
Výpis 37.6:	Definice třídy C37c v modulu m37c_Atributy.....	542
Výpis 37.7:	Ukázky reakcí na přístup k atributům ve třídě C37c	543
Výpis 37.8:	Definice třídy C37d v modulu m37c_Atributy.....	546
Výpis 37.9:	Práce s atributy instancí třídy C37d	547
Výpis 37.10:	Definice modulu m37d_ModuleProperty s vlastní mateřskou třídou	548
Výpis 37.11:	Import modulu m37d_ModuleProperty a práce s jeho vlastností	549
Výpis 37.12:	Ukázka používání slotů.....	551
Výpis 38.1:	Použití metody <code>__init_subclass__()</code>	553
Výpis 38.2:	Šablona testu funkce metatřídy v modulu m38a_TestMetaXxx	559
Výpis 38.3:	Import modulu m38a_TestMetaXxx.....	559
Výpis 38.4:	Definice metatřídy MetaFce definované jako funkce v modulu m38b_MetaFce.....	560
Výpis 38.5:	Import modulu m38b_MetaFce.....	561
Výpis 38.6:	Definice metatřídy MetaCls definované jako třída v modulu m38c_MetaCls	561
Výpis 38.7:	Import modulu m38c_MetaCls	562
Výpis 38.8:	Definice metatřídy MetaObj definované v modulu m38d_MetaObj jako objekt.....	563
Výpis 38.9:	Import modulu m38d_MetaObj	564
Výpis 38.10:	Definice jedináčka pomocí <code>__new__()</code>	565
Výpis 38.11:	Definice jedináčka pomocí metatřídy	566
Výpis 39.1:	Definice a použití jednoduchých korutin.....	570
Výpis 39.2:	Zabalení korutiny do instance třídy Task.....	571
Výpis 39.3:	Analýza datových typů	573

Seznam obrázků

Obrázek 1.1: Okno s dokumentací platformy	35
Obrázek 1.2: Okno příkazového řádku Windows se spuštěným interpretem Pythonu	39
Obrázek 1.3: Okno vývojového prostředí IDLE	41
Obrázek 1.4: Okno vývojového prostředí IDLE	42
Obrázek 4.1: Pokus o výpis rozsáhlejší nápovědy v prostředí IDLE	77
Obrázek 5.1: Indexování jednotlivých znaků textového řetězce – stringu	90
Obrázek 8.1: Syntaktický diagram příkazu <code>assert</code>	124
Obrázek 9.1: Okno světa želvy po provedení zadaných příkazů	135
Obrázek 9.2: Syntaktický diagram možných verzí příkazu <code>import</code>	141
Obrázek 12.1: Syntaktický diagram hlavičky funkce	179
Obrázek 13.1: Význam nastavení Show Code Context v prostředí IDLE	189
Obrázek 14.1: Postup zpracování jednoduchého podmíněného příkazu	204
Obrázek 14.2: Vývojový diagram úplného podmíněného příkazu	205
Obrázek 14.3: Vývojový diagram zobrazující postup vykonávání rozšířeného podmíněného příkazu	206
Obrázek 15.1: Syntaktický diagram příkazu <code>for</code>	221
Obrázek 16.1: Syntaktický diagram příkazu <code>try</code>	236
Obrázek 16.2: Syntaktický diagram příkazu vyhození výjimky	242
Obrázek 17.1: Syntaxe generátorové notace seznamu	250
Obrázek 22.1: Syntaktický diagram formátovacího stringu	315
Obrázek 22.2: Syntaktický diagram nahrazovacího pole	316
Obrázek 22.3: Syntaktický diagram nahrazovaného textu	316
Obrázek 22.4: Syntaktický diagram specifikace formátu	318
Obrázek 23.1: Indexování jednotlivých znaků textového řetězce – stringu	340
Obrázek 25.1: Diamantový problém	372
Obrázek 28.1: Architektura, pro kterou není možné odvodit MRO	415
Obrázek 31.1: Strom složek s balíčky a moduly	452
Obrázek 31.2: Stromy složek s NS-balíčky	465
Obrázek 34.1: Syntaktický diagram příkazu <code>with</code>	506
Obrázek 37.1: Reakce IDLE na zápis tečky za odkazem na objekt	544

Seznam tabulek

Tabulka 5.1: Tabulka priorit operátorů (operátory, u nichž nejsou uvedeny operandy, jsou binární, infixové).....	84
Tabulka 6.1: Klíčová slova jazyka Python.....	94
Tabulka 23.1: Tabulka zařazení jednotlivých probraných kontejnerů	337
Tabulka 23.2: Tabulka různých způsobů vykrajování	340
Tabulka 24.1: Důležité atributy a funkce modulu os	347
Tabulka 24.2: Probrané funkce modulu os.path	348
Tabulka 24.3: Režimy otevírání souboru	354
Tabulka 34.1: Tabulka přetížitelných binárních operátorů	499
Tabulka 34.2: Tabulka přetížitelných unárních operátorů	500
Tabulka 34.3: Tabulka zbylých emulačních funkcí	500
Tabulka 34.4: Tabulka konverzních funkcí	500

Seznam odboček – podšeděných bloků

Přísné a benevolentní programovací jazyky	49
Operátory jako funkční objekty	85
Statické a dynamické typování	93
Návrh podle kontraktu	124
Fyzické a logické řádky	128
Zásobník návratových adres – ZNA	213
Hešovatelné objekty	269
Předání argumentu odkazem	299
Mělké a hluboké kopie objektů	335
Terminologická vsuvka	365
Destruktor versus finalizér	496
Slabé odkazy (weak references)	550