

Stručný obsah

Část I	Základy	27
1	Startujeme.....	28
2	Zadávání jednoduchých hodnot	37
3	Jednoduché výrazy	49
4	Proměnné.....	72
5	Logické hodnoty a operace	86
6	Jednoduché příkazy	94
7	Moduly.....	103
8	Definice funkcí.....	126
9	Pokročilé rysy funkcí	146
10	Rozhodování a cykly	159
11	Ošetřování chyb	182
Část II	Kontejnery	195
12	Seznamy.....	196
13	N-tice	211
14	Množiny	222
15	Slovníky.....	233
16	Rozšíření definic funkcí	245
17	Operace se stringy a formátování.....	257
18	Operace s kontejnery	276
19	Práce se soubory	286

Část III	Objektově orientované programování	301
20	Základy OOP	302
21	Třídy a jejich instance	313
22	Jednoduché dědění.....	333
23	Násobné dědění.....	344
24	Přetěžování operátorů	355
25	Dekorátory	374
26	Deskriptory, vlastnosti a sloty	384
27	Balíčky.....	397
28	Ovlivnění tvorby tříd, metatřídy	401
29	Korutiny, vlákna, procesy.....	413
Část IV	Přílohy	421
A	Substituované disky ve <i>Windows</i>	422
B	Konvence pro psaní programů v <i>Pythonu</i>	424
	Literatura.....	427
	Rejstřík.....	428

Podrobný obsah

1	Startujeme.....	28
1.1	Hlavní součást instalace.....	28
Platforma.....		28
Dokumentace.....		29
Pracovní režimy.....		30
Vývojové prostředí.....		30
1.2	IDLE – seznamte se.....	31
Příkazové okno.....		32
Restart interaktivního systému.....		32
Uložení záznamu seance.....		33
Umístění editovaných souborů.....		33
Barevné zvýraznění textu.....		33
1.3	Objekty a objektové programování.....	34
Objekt, třída, instance.....		34
Objekt.....		34
Třída.....		34
Instance.....		34
1.4	Nejdůležitější zvláštnosti <i>Pythonu</i>	35
Přísné a benevolentní programovací jazyky.....		36
2	Zadávání jednoduchých hodnot.....	37
2.1	Zápis celých a desetinných (reálných) čísel.....	37
Zpřehlednění dlouhých čísel pomocí znaku podtržení.....		38
2.2	Komplexní čísla.....	39
2.3	Počáteční nula.....	40
2.4	Zadávání čísel v jiných číselných soustavách.....	40
2.5	Platí – neplatí.....	41
2.6	Zadávání textů.....	41
Escape sekvence.....		43
Bílé znaky.....		45
2.7	Nic – None.....	45
2.8	Výpustka – Ellipsis,	45
2.9	Literály.....	46
2.10	Důležitost přehlednosti.....	46
2.11	Komentáře.....	47
3	Jednoduché výrazy.....	49
3.1	Trocha teorie.....	49
Operace.....		49
Operand.....		49
Operátor.....		50

Arita operátorů	50
Priorita operátorů.....	51
Asociativita binárních operátorů.....	51
3.2 Volání funkčního objektu	53
Parametr versus argument	53
Syntaxe volání a návratová hodnota.....	54
Pořadí vyhodnocování a předávání argumentů.....	54
Datové × funkční objekty.....	55
Operátory jako funkční objekty	55
3.3 Zabudované funkce s jednoduchými argumenty	55
Zápis syntaxe	56
Přehled	56
abs(x).....	56
ascii(objekt).....	56
bin(x).....	56
bool(x)	56
complex([real[, imag]])	56
divmod(a, b).....	57
exec(object).....	57
float([x])	57
hash(object).....	57
help([objekt])	57
hex(i).....	57
id(object)	58
chr(i).....	58
input([výzva])	58
int([x]).....	58
int(s, base).....	58
len(s).....	59
max(arg1, arg2 ...) min(arg1, arg2 ...)	59
oct(i).....	59
ord(c).....	59
pow(x, y[, z]).....	59
print(argumenty)	59
range(stop) range(start, stop[, step])	60
repr(object).....	60
round(number[, ndigits]).....	60
str([object])	61
type(object)	61
3.4 Získání nápovědy.....	62
Argument zadán.....	62
Nápověda k některým operátorům a konstrukcím jazyka	63
Bez argumentu	63
3.5 Numerické operace.....	63
Tři druhy dělení.....	64
Umocňování.....	65
Nekonečna a nesmyslná čísla	66
3.6 Bitové posuny	67
3.7 Operace s texty	68
Sčítání textů.....	68
Násobení textů	68
Indexace jednotlivých znaků	69
3.8 Zadání na více řádcích	70

4	Proměnné	72
4.1	Co jsou to proměnné	72
	Statické a dynamické typování	73
4.2	Správa paměti	74
4.3	Pravidla pro tvorbu identifikátorů	74
4.4	Zavedení proměnné	75
	Zadání skupiny hodnot	76
	Proměnné inf a nan	78
4.5	Vnořená volání funkčních objektů	78
4.6	Zjištění typu objektu v proměnné	80
4.7	Uložení funkce do proměnné	81
4.8	Lambda-výrazy	82
	Vnořování volání funkčních objektů	82
4.9	Datové a funkční proměnné	83
4.10	Mezery ve výrazech a příkazech	84
4.11	Uložení do proměnné x propojení s názvem	84
4.12	Přiřazovací výraz	84
5	Logické hodnoty a operace	86
5.1	Konstanty True a False	86
5.2	Převod jiných hodnot na logické	87
5.3	Porovnávání hodnot	88
	Porovnání reálných čísel	88
	Porovnávání textů	88
	Zřetěžené porovnávání	89
	Porovnávání totožnosti objektů	89
5.4	Logické operátory a operace	90
5.5	Operace s jednotlivými bity	91
5.6	Podmíněný výraz	92
6	Jednoduché příkazy	94
6.1	Příkaz pass	94
6.2	Příkaz tvořený výrazem, výrazový příkaz	94
6.3	Několik příkazů na řádku	95
6.4	Přiřazovací příkaz	95
	Složená přiřazení	95
6.5	Příkaz del	96
6.6	Příkaz assert	97
	Návrh podle kontraktu	98
6.7	Složené příkazy a odsazování	100
	Výhody a nevýhody koncepce <i>Pythonu</i>	101
7	Moduly	103
7.1	Trocha teorie – základní informace	103
	Atributy a metody	103
	Práce s objekty – kvalifikace	104
	Vše je součástí nějakého modulu	104
	Dva názvy objektů	105
	Zdrojový soubor	105
	Přeložený soubor	105
7.2	Příkaz import	106
	Čistý import jiného modulu	106
	Import modulu pod jiným názvem	107
	Přímý import vyjmenovaných objektů	109

Import objektů modulu nezahrnuje import modulu.....	110
Argumentem příkazů import a from ... import musí být modul.....	110
Import všech objektů.....	110
Syntaktické diagramy příkazu import.....	111
7.3 Modul jako objekt.....	112
Modul builtins a zdánlivě neobjektové programování.....	112
7.4 Postup systému při importu modulu.....	113
Prohledávané složky.....	113
7.5 Vytvoření vlastního modulu.....	114
Kódová stránka.....	116
Dokumentační komentář.....	116
Zadané příkazy.....	116
Informace o načítání modulu.....	117
Definice datových atributů.....	117
Neveřejné atributy.....	117
Výrazové příkazy.....	118
7.6 Průběh importu vytvořeného modulu.....	118
Použitelné názvy.....	119
Neveřejné atributy.....	120
7.7 Reimport již importovaného modulu.....	121
Přímo importované proměnné si zachovávají své odkazy.....	122
Specifika funkce importlib.reload().....	123
Rozbor chybového hlášení.....	123
Důsledky chybného zavedení modulu.....	124
Syntaktické chyby.....	124
7.8 Zprostředkovaný import.....	125
8 Definice funkcí.....	126
8.1 Definice funkce.....	127
8.2 Definice vlastní funkce.....	127
8.3 Kdy se projeví chyby v definici funkce.....	130
8.4 Parametry, argumenty a lokální proměnné.....	131
Definice.....	132
Lokální proměnné.....	132
Volání funkcí s parametry.....	133
Povinně pojmenované argumenty.....	134
Povinně poziční argumenty.....	135
Mix povinností.....	136
8.5 Implicitní hodnoty argumentů.....	138
8.6 Konstantnost předdefinovaných hodnot.....	139
8.7 Funkce s vedlejším efektem.....	140
8.8 Funkce vracející hodnotu a příkaz return.....	141
8.9 Přetěžování funkcí.....	142
Něco přetížít jde.....	143
8.10 Použití anotací.....	143
8.11 Definice funkcí v modulu.....	145
9 Pokročilé rysy funkcí.....	146
9.1 Vnitřní funkce.....	146
Odsazování.....	147
9.2 Jmenné prostory a oblast platnosti.....	147
9.3 Oblast/rozsah platnosti, působnost (scope).....	148
Zanoření jmenových prostorů.....	149
9.4 Lokalita použitých proměnných.....	149
Příkaz global.....	151

Příkaz <code>nonlocal</code>	152
9.5 Vnoření funkce versus vnoření volání funkcí	154
9.6 Vnořená volání funkcí	154
9.7 Funkce vyššího řádu	155
9.8 Atributy funkcí	157
9.9 Nelokální proměnné a uzávěry (closures)	157
9.10 Další vlastnosti funkcí	158
10 Rozhodování a cykly	159
10.1 Podmíněný příkaz	159
Jednoduchý podmíněný příkaz	160
Větev <code>else</code>	160
Rozhodování s více větvemi: rozšířený podmíněný příkaz	162
10.2 Rekurze	162
Zásobník návratových adres – ZNA	165
10.3 Příkaz <code>while</code> – cyklus se vstupní podmínkou	165
10.4 Nekonečný cyklus	166
10.5 Příkaz <code>break</code>	167
10.6 Cyklus s ukončovací podmínkou	168
10.7 Přiřazení v hlavičce cyklu	168
10.8 Větev <code>else</code>	170
10.9 Příkaz <code>continue</code>	170
10.10 Cyklus <code>for</code>	172
Vyjmenování hodnot parametru cyklu	172
Využití funkce <code>range()</code>	173
Použití indexů	174
Použití stringu jako zdroje	174
Vnořování cyklů	175
Postupné použití několika zdrojů	175
Definice generátorové funkce jako zdroje	176
Operátor <code>yield</code> jako výraz	177
Výraz/příkaz <code>yield from</code>	177
Větev <code>else</code>	179
10.11 Problémy s odsazením v interaktivním režimu	179
10.12 Modul s několika užitečnými funkcemi	181
11 Ošetřování chyb	182
11.1 Tři druhy chyb	182
Syntaktické chyby	182
Běhové chyby	182
Logické chyby	183
11.2 Chybové zprávy	183
Syntaktické chyby při interpretaci příkazu v interaktivním režimu	184
Syntaktické chyby odhalené při jiné příležitosti	184
Běhové chyby	185
11.3 I chyby jsou objekty – výjimky	187
11.4 Rozdělení výjimek	187
11.5 Zachycení a ošetření výjimky	188
11.6 Více větví <code>except</code>	189
11.7 Větev <code>else</code>	189
11.8 Větev <code>finally</code>	190
11.9 Syntaktický diagram příkazu <code>try</code>	190
11.10 Příklad s kompletní verzí příkazu <code>try</code>	191

Převod se nepodařil	191
Převod se podařil	191
11.11 Praktický příklad	193
11.12 Vyhození vlastní výjimky	193
11.13 Příkaz assert	194
11.14 Hierarchie výjimek a definice vlastní výjimky	194

Část II Kontejnery 195

12 Seznamy	196
12.1 Proměnné a neměnné objekty	196
12.2 Základní informace o seznamech	197
12.3 Vytváření seznamů	198
Použití literálu	198
Využití funkce list(seq=())	199
Sčítání a násobení	200
12.4 Generátorová notace seznamů	200
12.5 Modifikace seznamů	202
Metody append() a extend()	202
Rizika práce s odkazy na proměnné objekty	203
Postupné budování seznamu	204
Přičítání jiných zdrojů	205
Indexace prvků seznamu	206
Metody pracující s indexy	206
index(value, start=0, stop=9223372036854775807, /) -> int	206
insert(index, object, /)	207
pop(index=-1, /) -> ?	207
remove(self, value, /)	207
Metody pracující s celým seznamem	208
reverse(self, /)	208
sort(*, key=None, reverse=False)	208
12.6 Vícerozměrné seznamy	208
12.7 Souhrnný příklad	209
13 N-tice	211
13.1 Základní informace o n-ticích	211
13.2 Vytváření n-tic	212
Vytváření n-tic pomocí literálů	212
Využití funkce tuple(seq=())	213
Sčítání a násobení	214
Přičítání n-tic	215
Balení a rozbalování n-tic	216
Prohazování proměnných	217
Hvězdičkové pravidlo	217
13.3 Generátorová notace n-tic	218
13.4 Problematika neměnnosti n-tic	219
Hešovatelné objekty	219
13.5 Přístup k prvkům n-tic	220
13.6 Proměnné a neměnné prvky n-tice	220
13.7 Sčítání seznamů a n-tic	221
14 Množiny	222
14.1 Základní informace o množinách	222

14.2	Vytváření množin	222
	Vytváření množin pomocí literálů	223
	Vytváření množin pomocí funkce <code>set()</code>	223
	Hešová tabulka	223
	Další pužitelné zdroje	224
	Vytváření množin prostřednictvím množinových operací	224
	<code>union(*zdroj) a b ...</code>	225
	<code>intersection(*zdroj) a & b & ...</code>	225
	<code>difference(*zdroj) a - b - ...</code>	225
	<code>symmetric_difference(zdroj) a ^ b</code>	225
14.3	Generátorová notace množin	227
14.4	Zmrazené množiny	227
14.5	Modifikace množin	228
	Modifikace pracující s jedním prvkem	228
	<code>add(element)</code>	228
	<code>discard(element)</code>	228
	<code>remove(element)</code>	228
	<code>pop()</code>	228
	Množinové operátory a sdružené operace	229
	<code>update(*zdroj) a = b ...</code>	230
	<code>intersection_update(*zdroj) a &= b & ...</code>	230
	<code>difference_update(*zdroj) a -= b ...</code>	231
	<code>symmetric_difference_update(zdroj) a ^= b</code>	231
	Porovnávání množin	231
	<code>isdisjoint(množina)</code>	231
	<code>issubset(množina) a <= b</code>	231
	<code>a < b</code>	231
	<code>issuperset (množina) a >= b</code>	231
	<code>a > b</code>	232
15	Slovníky	233
15.1	Mapovací objekty a slovníky	233
15.2	Vytváření slovníků	234
	Vytváření slovníků pomocí literálů	234
	Vytváření slovníků pomocí funkce <code>dict()</code>	235
	Ekvivalence slovníků	236
	Vytváření slovníků pomocí metody <code>fromkeys()</code>	236
15.3	Generátorová notace slovníků	237
15.4	Operace se slovníkem	238
	Práce s hodnotami pomocí „indexace“ klíčem	238
	Další metody pro práci s jednotlivými položkami	239
	<code>get(key, default=None, \)</code>	240
	<code>pop(key, default=#, \)</code>	240
	<code>popitem()</code>	240
	<code>setdefault(key, default=None, \)</code>	240
	Modifikace slovníku daty ze zadaného zdroje	241
	<code>update(zdroj)</code>	241
15.5	Pohledy	241
	<code>items()</code>	241
	<code>keys()</code>	241
	<code>values()</code>	241
	Pohledy jako generátory	242
	Operace s pohledy	243

16 Rozšíření definic funkcí	245
16.1 Předávání argumentů odkazem a hodnotou	245
Předání argumentu odkazem.....	246
16.2 Proměnný počet pozičních argumentů	247
Hvězdičkový parametr.....	247
Hvězdičkový argument.....	249
16.3 Proměnný počet pojmenovaných argumentů.....	250
Dvouhvězdičkový parametr	250
Dvouhvězdičkový argument	251
16.4 Stručný souhrn	252
16.5 Zabudované funkce pracující s kontejnery	252
all(iterable).....	252
any(iterable).....	253
dir([object])	253
enumerate(iterable, start=0).....	253
eval(expression, globals=None, locals=None)	254
filter(function, iterable)	254
globals()	254
len(c).....	254
locals()	255
max(iterable, *, [key, default]).....	255
max(arg1, arg2, *args[, key])	255
min(iterable, *, [key, default]).....	255
min(arg1, arg2, *args[, key])	255
reversed(seq).....	255
slice(stop) slice(start, stop[, step]).....	256
sorted(iterable, *, key=None, reverse=False).....	256
sum(iterable, /, start=0)	256
vars([object])	256
zip(*iterables)	256
17 Operace se stringy a formátování	257
17.1 Prefixy stringových literálů.....	257
17.2 Formátovací operátor %.....	258
17.3 Metoda format()	260
Formátovací string.....	260
Konverze.....	262
Specifikace formátu	262
Počet zabraných pozic.....	263
Přesnost.....	263
Typ	264
Skupiny	266
Alternativní formát a vedoucí nuly.....	267
Znaménko	268
Zarovnání a pinění	268
Vnořená nahrazovací pole	269
17.4 Stringová interpolace – f-stringy.....	270
17.5 Příklad: Pascalův trojúhelník.....	272
17.6 Příklad: Trasovací funkce	273
18 Operace s kontejnery	276
18.1 Proměnné objekty jako implicitní hodnoty parametrů.....	276
18.2 Kopírování	276
Mělké a hluboké kopie objektů.....	277

Alternativní způsob tvorby mělkých kopií	278
Nebezpečí hlubokých kopií	278
18.3 Rozdělení doposud probraných kontejnerů	279
18.4 Řazení prvků posloupnosti.....	280
reversed(seq)	280
sorted(iterable, *, key=None, reverse=False).....	280
18.5 Vykrajování (slicing).....	281
18.6 Indexování a vykrajování u rozsahů	282
18.7 Nahrazování hodnot	283
18.8 „Úprava“ neměnných objektů	285
19 Práce se soubory	286
19.1 Soubory: bleskové opakování.....	286
Soubor, souborový systém, cesta	286
Absolutní a relativní cesta	287
Substituované disky ve Windows.....	288
19.2 Dva způsoby práce se souborovým systémem.....	288
19.3 Moduly os a os.path	288
19.4 Pracovní složka.....	291
19.5 Skládání a rozkládání cest	292
19.6 Získání informací o souborech.....	293
19.7 Vytváření složek.....	294
19.8 Zápis a čtení dat.....	295
Otevírání souborů.....	295
Zápis dat, splachování a zavírání souborů.....	297
19.9 Čtení ze souborů.....	298

Část III Objektově orientované programování 301

20 Základy OOP	302
20.1 Základní princip OOP	303
20.2 Objekty a jejich atributy	304
20.3 Třídy a jejich instance	305
Třída versus datový typ	305
Instance	305
20.4 Objekt třídy versus instance třídy	306
20.5 Atributy třídy versus atributy instancí.....	306
20.6 Zprávy	307
20.7 Metody	307
20.8 Dědění.....	308
Terminologie	308
LSP – substituční princip Liskové.....	308
Virtuální metody a jejich přebíjení	309
Zakrývání versus přebíjení	309
Polymorfismus.....	310
Rodičovský podobjekt	310
Násobné dědění a diamantový problém	310
Zobecňování	312
21 Třídy a jejich instance	313
21.1 Definice třídy a jejich atributů	313
Definice třídy je příkaz	314
21.2 Práce s atributy objektu	315

Získání a modifikace hodnoty atributu	315
Přidání a odebrání atributu	316
Třída jako parametr	317
21.3 Instance a práce s nimi	317
Tři druhy metod	319
Metody instancí a parametr <code>self</code>	319
Vytváření instancí – konstruktor	320
Instanční metody třídy C2	321
Použití instančních a třídních atributů	323
Změny instančních a třídních datových atributů	324
Zavádění nových třídních funkčních atributů	327
Zavádění nových instančních funkčních atributů	328
21.4 Zabudované třídy jsou nemodifikovatelné	328
21.5 Nutnost kvalifikace atributů	329
21.6 Speciální atributy	330
<code>__bases__</code>	330
<code>__class__</code>	330
<code>__dict__</code>	330
<code>__doc__</code>	330
<code>__mro__</code>	330
<code>__name__</code>	330
<code>__qualname__</code>	331
<code>__subclasses__()</code>	331
Ukázky použití	331
21.7 Zabudované třídy	332
22 Jednoduché dědění	333
22.1 Vytváření potomka a jeho vlastnosti	333
22.2 Příklad	334
Třída LA	334
Třídy LB a LC	336
22.3 Jmenné prostory	336
22.4 Používání třídních atributů	337
22.5 Volání metod	338
22.6 Konstruktory se dědí	339
22.7 Hierarchie výjimek	340
22.8 Definice vlastní výjimky	340
22.9 Iterovatelné objekty a iterátory	341
22.10 Zabudované funkce pracující s objekty	342
<code>callable(object)</code>	342
<code>classmethod(method)</code>	342
<code>delattr(object, name)</code>	342
<code>getattr(object, name[, default])</code>	342
<code>hasattr(object, name)</code>	342
<code>isinstance(object, classinfo)</code>	342
<code>issubclass(class, classinfo)</code>	342
<code>iter(object[, sentinel])</code>	342
<code>map(function, iterable, ...)</code>	343
<code>next(iterator[, default])</code>	343
<code>setattr(object, name, value)</code>	343
<code>staticmethod</code>	343
<code>super([/type[, object-or-type]])</code>	343
23 Násobné dědění	344

23.1	Python a násobné dědění	344
23.2	Pomocná funkce <code>prsk()</code> v modulu <code>utils</code>	345
23.3	Příklad: jednoduchý diamant	345
23.4	Analýza chování	347
	Přímé zadání volaného konstrukturu	348
23.5	Virtuální metody	350
23.6	Složitější hierarchie dědění a MRO	350
	Nerealizovatelná a následně opravená hierarchie	350
	Zásady specifikace MRO	351
23.7	Komolení jmen a pseudosoukromé atributy	352
23.8	Šablonová metoda	353
24	Přetěžování operátorů	355
24.1	Základy	355
	Neimplementované operace – <code>NotImplemented</code>	356
24.2	Základní sada	356
	<code>__bool__(self)</code>	356
	<code>__del__()</code>	356
	<code>__lt__(self, other)</code> <code>__le__(self, other)</code> <code>__eq__(self, other)</code>	
	<code>__gt__(self, other)</code> <code>__ge__(self, other)</code>	356
	<code>__hash__(self)</code>	357
	<code>__format__(self, format_spec)</code>	357
	<code>__repr__(self)</code> , <code>__str__(self)</code>	357
24.3	Operátory <code>+</code> <code>-</code> <code>*</code> <code>@</code> <code>/</code> <code>//</code> <code>%</code> <code>**</code> <code><<</code> <code>>></code> <code>&</code> <code>^</code> <code> </code> a emulace numerických typů	357
	Binární operátory	358
	Unární operátory	359
	Zbylé emulační funkce	359
	Konverzní funkce	359
	Souhrnný příklad	360
24.4	Operátor <code>[]</code> a emulace kontejneru	362
	<code>__contains__(self, item)</code>	362
	<code>__delitem__(self, key)</code>	362
	<code>__getitem__(self, key)</code>	362
	<code>__iter__(self)</code>	362
	<code>__len__(self)</code>	362
	<code>__missing__(self, key)</code>	362
	<code>__reversed__(self)</code>	363
	<code>__setitem__(self, key, value)</code>	363
	Příklad	363
24.5	Operátor <code>()</code> – volatelné objekty	363
	Příklad	364
24.6	Operátor <code>.</code> (tečka) – přístup k atributům	365
	<code>__getattr__(self, name)</code>	365
	<code>__getattribute__(self, name)</code>	365
	<code>__setattr__(self, name, value)</code>	366
	<code>__delattr__(self, name)</code>	366
	<code>__dir__(self)</code>	366
	Příklad – definice	366
	Příklad – kontrola	366
	Aktivita IDLE	368
	Další příkazy	369
	Lepší řešení	370
24.7	Správce kontextu a příkaz <code>with</code>	372
	Příklad	373

25 Dekorátory	374
25.1 Co jsou dekorátory	374
„Operátor“ @	375
25.2 Použití dekorátorů staticmethod a classmethod	375
Instanční versus třídní versus statické metody	375
Příklad použití	376
Příklad s třídní metodou	377
25.3 Tvorba vlastního dekorátoru	378
25.4 Dekorátory třídy	380
Jedináček – singleton	380
25.5 Dekorátory s parametry	383
25.6 Současné použití více dekorátorů	383
26 Deskriptory, vlastnosti a sloty	384
26.1 Co jsou deskriptory	384
__get__(self, instance, owner=None)	384
__set__(self, instance, value)	384
__delete__(self, instance)	385
__set_name__(self, owner, name)	385
Dělení deskriptorů	385
26.2 Příklad	385
26.3 Deskriptor vytvářený vnitřní třídou	388
26.4 Příklad: deskriptor pro odložené vyhodnocení	389
26.5 Co jsou to vlastnosti	390
Přímá definice vlastnosti	390
Definice ekvivalentu třídy property	392
26.6 Zadání vlastnosti pomocí dekorátoru	393
26.7 Jaký způsob správy přístupu k atributu zvolit	393
26.8 Sloty	395
Slabé odkazy (weak references)	395
27 Balíčky	397
27.1 Balíčky (packages)	397
Konstruktor balíčku	397
Implementace	398
Úpravy modulu v balíčku a jeho opětovné načtení	399
Nedostupnost neimportovaných balíčků	400
27.2 Vytvoření aplikace či knihovny	400
28 Ovlivnění tvorby tříd, metatřídy	401
28.1 Abstraktní třídy a abstraktní metody	401
Proč definovat metodu použitou v šabloně	403
Proč dekorovat metodu jako abstraktní	403
Proč v hlavičce deklarovat metatřídu	403
Shrnutí	403
28.2 Ovlivnění tvorby dceřiných tříd	404
Zákaz definice potomků dané třídy	404
28.3 Co jsou to metatřídy	405
28.4 Postup při vytváření třídy	405
28.5 Různé definice metatříd	406
Vzorový kód pro demonstraci funkce metatřídy	406
Metatřída definovaná jako funkce	408
Metatřída definovaná jako třída	409
Metatřída definovaná jako objekt	411
28.6 Shrnutí	412

29 Korutiny, vlákna, procesy	413
29.1 Paralelní provádění více činností.....	413
Kooperativní plánování.....	413
Preemptivní plánování.....	414
Procesy – vlákna – koprogramy.....	414
29.2 Koprogramy, korutiny a očekávatelné objekty.....	414
29.3 Knihovna/modul <code>asyncio</code>	415
29.4 Definice a použití korutin.....	416
Využití objektů typu <code>Task</code>	417
Korutiny jsou instancemi třídy <code>coroutine</code>	418
29.5 Asynchronní cyklus – příkaz <code>async for</code>	419
29.6 Asynchronní správce kontextu – <code>async with</code>	419
29.7 Práce s vlákny.....	420
29.8 Práce s procesy.....	420

Část IV Přílohy 421

A Substituované disky ve <i>Windows</i>	422
B Konvence pro psaní programů v <i>Pythonu</i>	424
Uspořádání kódu.....	424
Jmenné konvence.....	425
Dokumentační komentáře (PEP 257).....	426
Literatura	427
Rejstřík	428